

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»  
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ  
*Кафедра автоматизованих систем обробки інформації і управління*

До захисту допущено:

В.о. завідувача кафедри

\_\_\_\_\_ Олександр ПАВЛОВ  
(підпис) (вл.ім'я, прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2020 р.

**Дипломний проєкт**  
**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Інформаційні управляючі  
системи та технології»  
спеціальності 122 «Комп'ютерні науки та інформаційні технології»**

**на тему: « Інформаційно-аналітична система класифікації  
архетипу персонажа »**

**Виконав:**

студент IV курсу, групи ІС-63

\_\_\_\_\_ Восієв Руслан Зійодуллович

(прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

**Керівник**

\_\_\_\_\_ доц., к.т.н., доц. Фіногенов Олексій Дмитрович

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

**Консультант з  
графічної  
документації**

\_\_\_\_\_ ст. вик. Проскура Світлана Леонідівна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

**Рецензент**

\_\_\_\_\_ доц., к.т.н., доц. Мураховський Сергій

Анатолійович

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

Засвідчую, що у цьому дипломному проєкті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2020 року

**Національний технічний університет України**  
**“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки  
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління  
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

\_\_\_\_\_ Олександр ПАВЛОВ  
(підпис) (вл.ім'я, прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2020 р.

**ЗАВДАННЯ**  
**на дипломний проєкт студенту**

Восієву Руслану Зійодуловичу  
(прізвище, ім'я, по батькові)

1. Тема проєкту « Інформаційно-аналітична система класифікації архетипу персонажа »

керівник проєкту Фіногенов Олексій Дмитрович, к.т.н., доцент  
( прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “7” травня 2020 р. №1081-с

2. Термін подання студентом проєкту “05” червня 2020 року

3. Вихідні дані до проєкту

*Технічне завдання*

4. Зміст пояснювальної записки

*1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі*

*2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних*

*3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання*

*4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення, побудова звітів*

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

6. Перелік графічного матеріалу

1. Схема структурна варіантів використання

2. Схема структурна діяльності

3. Схема бази даних

4. Схема структурна класів програмного забезпечення

5. Схема структурна послідовності

6. Схема структурна компонентів програмного забезпечення

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «13» квітня 2020 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення рекомендованої літератури	14.04.2020	
2.	Аналіз існуючих методів розв'язання задачі	16.04.2020	
3.	Постановка та формалізація задачі	18.04.2020	
4.	Розробка інформаційного забезпечення	22.04.2020	
5.	Алгоритмізація задачі	26.04.2020	
6.	Обґрунтування використовуваних технічних засобів	27.04.2020	
7.	Розробка програмного забезпечення	29.04.2020	
8.	Налагодження програми	09.05.2020	
9.	Виконання графічних документів	10.05.2020	
10.	Оформлення пояснювальної записки	11.05.2020	
11.	Подання ДП на попередній захист	15.05.2020	
12.	Подання ДП на основний захист	06.06.2020	
13.	Подання ДП рецензенту	07.06.2020	

Студент

Руслан ВОСІЄВ

Керівник

Олексій ФІНОГЕНОВ

[illegible]

# **Пояснювальна записка до дипломного проєкту**

на тему: Інформаційно-аналітична система класифікації архетипу  
персонажа

---

Київ – 2020 року

## АНОТАЦІЯ

**Структура та обсяг роботи.** Пояснювальна записка дипломного проекту складається з п'ятих розділів і містить 60 сторінок, 17 рисунків, 7 таблиць, 1 додаток, 15 джерел.

У дипломному проекті реалізовано інформаційно-аналітичну систему класифікації архетипу персонажа. Метою системи є спрощення процесу літературного аналізу творів та дослідження залежності архетипів персонажів літературних творів від певних факторів. Задачами розробки є тренування моделі на різних вибірках даних, визначення архетипу введеного персонажа у певному творі, отримання описової характеристики кожного архетипу, отримання історії використання програми. Користувач системи матиме можливість провести класифікацію власного тексту для визначення архетипу обраного персонажа.

У розділі загальних положень описано основні цілі класифікації тексту як напряму у сфері машинного навчання і визначено необхідність розробки поглибленого аналізу літературних творів як для розвитку сфери в цілому, так і для прикладного застосування системи.

У розділі інформаційного забезпечення виконано планування та побудова структури бази даних, описано сутності у вигляді таблиць, наведено графічне зображення структури бази даних.

У розділі математичного забезпечення описано змістовну та математичну постановки, наведено алгоритм розв'язання поставленої задачі.

У розділі програмного та технічного забезпечення описана архітектура системи, вказано засоби розробки, визначено вимоги до технічного забезпечення сервера та клієнта, описано архітектуру системи. Наведено

					ДП 6306.00.000 ПЗ						
Зм.	Арк.	Прізвище	Підпис	Дата	Інформаційно-аналітична система класифікації архетипу персонажа	Літ.		Арк.		Аркушів	
Розроб.		Восієв Р.З.						5		60	
Перевірив.		Фіногенов О.Д.				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63					
Н. кон.		Проскура С.Л.									
Затв.		Павлов О.А.									

детальний опис класів та функцій програмного продукту, побудовано UML-діаграми.

У технологічному розділі описано керівництво користувача на прикладі навігації системою та графічного зображення роботи системи. Проведено тестування функціональних вимог методом випробувань.

КЛАСИФІКАЦІЯ ТЕКСТІВ, МЕТОД ОПОРНИХ ВЕКТОРІВ,  
КЛАСИФІКАЦІЯ АРХЕТИПУ ПЕРСОНАЖА, ВЕБ-ЗАСТОСУНОК

					ДП 6306.00.000 ПЗ	Авк.
Змн.	Арк.	№ докум.	Підпис	Дата		

## ABSTRACT

**The structure and scope of this paper.** Explanatory note of the graduation project consists of five sections, contains 60 pages, 17 figures, 7 tables, 1 appendice, 15 references.

The graduation project has been implemented as an informational-analytical system of classification of the character archetype. The goal of a system is making the process of literary analysis of works and the progress of deposits of archetypes of characters in literary works of singing factors easier. The tasks of development is training models on different datasets, recognizing the archetype of the entered character in the literary composition, receiving the descriptive characteristics of the character archetypes, and displaying the history of the program from the past. System's user will be able to carry out the classification of the text for the recognition of the archetype of the character.

The general provisions section describes the main objectives of text classification as a direction in the field of machine learning and identifies the need to develop an in-depth analysis of literary works for the development of the field as a whole and for the application of the system.

In the section of information support the planning and construction of the database structure is performed, the essences are described in the form of table, the graphic image of the database structure is given. The distribution of mathematical security describes the mathematical setting and the algorithm of the development of the assigned tasks.

The software and hardware section describes the system architecture, indicates the development tools, defines the requirements for server and client hardware, describes the system architecture. A detailed description of the class and functionality of a software product, is prompted by UML diagrams. Listed detailed description of classes and functions of the software product, built UML-diagrams.

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		



The technological section describes the user manual on the example of system navigation and graphical representation of the system. Testing of functional requirements by a test method is carried out.

TEXTS CLASSIFICATION, SUPPORT VECTOR MACHINE,  
CHARACTER ARCHETYPE CLASSIFICATION, WEB-APPLICATION

					ДП 6306.00.000 ПЗ	Авк.
Змн.	Арк.	№ докум.	Підпис	Дата		

## ЗМІСТ

ВСТУП.....	11
<b>1 ЗАГАЛЬНІ ПОЛОЖЕННЯ .....</b>	<b>12</b>
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА .....	12
1.1.1 Опис процесу діяльності.....	12
1.1.2 Опис функціональної моделі.....	13
1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ .....	17
1.3 ПОСТАНОВКА ЗАДАЧІ.....	18
1.3.1 Призначення розробки.....	18
1.3.2 Цілі та задачі розробки .....	19
Висновок до розділу .....	19
<b>2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>20</b>
2.1 ВХІДНІ ДАНІ .....	20
2.2 ВИХІДНІ ДАНІ.....	20
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ .....	20
Висновок до розділу .....	23
<b>3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>24</b>
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ .....	24
3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ .....	25
3.3 ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ.....	25
3.4 ОПИС МЕТОДІВ РОЗВ'ЯЗАННЯ .....	27
Висновок до розділу .....	30
<b>4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>31</b>
4.1 ЗАСОБИ РОЗРОБКИ .....	31
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ .....	32
4.2.1 Загальні вимоги .....	32
4.3 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	33

4.3.1	Діаграма класів .....	37
4.3.2	Діаграма послідовності .....	39
4.3.3	Діаграма компонентів .....	41
4.3.4	Специфікація функцій .....	43
	Висновок до розділу .....	45
<b>5</b>	<b>ТЕХНОЛОГІЧНИЙ РОЗДІЛ .....</b>	<b>46</b>
5.1	КЕРІВНИЦТВО КОРИСТУВАЧА .....	46
5.1.1	Головне меню .....	46
5.1.2	Авторизація .....	47
5.1.3	Тренування.....	48
5.1.4	Класифікація .....	49
5.1.5	Результати класифікацій.....	51
5.1.6	Архетипи .....	52
5.1.7	Історія класифікацій .....	54
5.2	ВИПРОБУВАННЯ ПРОГРАМНОГО ПРОДУКТУ .....	56
5.2.1	Мета випробувань .....	56
5.2.2	Загальні положення .....	56
5.2.3	Результати випробувань .....	57
	Висновок до розділу .....	61
	ЗАГАЛЬНІ ВИСНОВКИ .....	62
	ПЕРЕЛІК ПОСИЛАНЬ .....	63
	ДОДАТОК А .....	65

## ВСТУП

Комп'ютерна лінгвістика як науковий напрям існує уже майже 70 років. Поява перших мов програмування, а також поступовий розвиток інформаційних технологій дав поштовх для розвитку машинного перекладу текстів, особливо радянських наукових журналів. Хоча і такий переклад значно поступався і поступається у якості «живому», розвиток напрямку є значним і заслуговує на окрему увагу.

У сучасному світі для повсякденного користувача інтернету не є новиною можливість перекладати текст онлайн, виконувати миттєвий пошук слів, або визначати автора на назву тексту за уривком. Більш того, за останні роки використання систем штучного інтелекту стало нормою для виконання прикладних лінгвістичних завдань. Одним з таких завдань є класифікація тексту.

Існує велика кількість параметрів класифікації тексту, як і методів автоматизації даної задачі шляхом програмування. Проте, якщо для глобальних завдань знайдено велику кількість можливих рішень, для більш детального аналізу тексту їх значно менше.

Класифікація архетипу персонажа є новою задачею комп'ютерної лінгвістики, адже досі не було створено відповідних методів її вирішення. Персонажі літературних творів здебільшого виступають у ролі шаблонних фігур з певними індивідуальними рисами і кожен з таких шаблонів має певні особливості, які відрізняють їх один від одного. Автоматизація визначення характерних ознак героїв шляхом програмування є ключовим завданням.

**Практичне значення одержаних результатів.** Розроблено алгоритм класифікації архетипу персонажа літературного твору.

					ДП 6306.00.000 ПЗ	Адк.
Змн.	Арк.	№ докум.	Підпис	Дата		

## 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

### 1.1 Опис предметного середовища

Предметним середовищем є процес класифікації текстів на основі інформації про окремо взятих персонажів з попереднім тренуванням моделі на певній вибірці даних.

Автором класифікації людей за архетипами є Карл Густав Юнг [1]. Він був психологом і його ідея полягала у тому, що кожна людина виступає в якості носія колективного несвідомого — того, що представляє собою спільні особливості для різних осіб.

На основі вчень Юнга створено велику кількість архетипів, проте наведена класифікація є основною — 5 типів, символізуючих людську мотивацію.

Список основних архетипів наведено нижче [2]:

- невинний — розвиток віри, впевненості й оптимізму;
- простак — усвідомлення існування поганих речей, розвиток реалізму;
- герой — навчання змагання, досягнення цілей і захист себе;
- наставник — демонстрація турботи і співчуття до інших, допомога;
- злодій — мета завадити героєві досягти своєї мети.

Даний розподіл використовується і для героїв літературних творів, адже вони здебільшого є прототипами реальних людей.

#### 1.1.1 Опис процесу діяльності

Для проведення літературного аналізу певного твору науковцеві-літературознавцю необхідно витратити велику кількість часу. Прочитання більшості літературних творів у звичайної людини займає десятки годин, якщо це роман чи повість. Тим більше, одного читання не достатньо для проведення

класифікації тексту та достовірного аналізу його змісту. Враховуючи тематику даного проекту, мова йде про аналіз придатності літературного персонажа до певного архетипу.

Автоматизація роботи полягає перш за все в економії часу на аналіз. По-друге, сучасні технології надають можливість виконувати завдання класифікації з точністю близько до 1, в залежності від розміру тренувальної вибірки.

Сам аналіз полягає у неодноразовому прочитанні твору літературознавцем, визначенні абзаців, що містять опис персонажа, його взаємодію з іншими дійовими особами твору, розвиток від початку розповіді до її розв'язки.

### 1.1.2 Опис функціональної моделі

Даний проект розроблюється для дослідницьких цілей, а саме аналізу роботи та використання алгоритмів класифікації для більш вузької спеціалізації, і передбачає взаємодію з одним користувачем, тому актор один.

					ДП 6306.00.000 ПЗ	Аж.
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 1.1 — Функціональні вимоги

Актор	Варіант використання	Функціональна вимога	Пріоритет
Користувач	Головне меню	<p>— Система надає користувачеві короткий опис роботи програми.</p> <p>— Система надає користувачеві можливість скористатись кнопкою «Почати роботу» для переходу на сторінку тренування моделі.</p> <p>— Система надає можливість користувачеві скористатись меню вгорі для швидкого переходу між сторінками у системі.</p>	Високий
Адміністратор	Головне меню	<p>— Система надає можливість авторизуватись адміністраторові за допомогою кнопки «Увійти».</p> <p>— Система надає можливість авторизованому користувачеві вийти з облікового запису за допомогою кнопки «Вийти».</p>	Високий

Таблиця 1.1 (продовження)

Користувач	Додавання датасету	<p>— Система надає можливість користувачеві обрати датасет для тренування моделі.</p> <p>— Система надає можливість користувачеві обирати датасети з різних жанрів літератури за допомогою випадального меню.</p>	Високий
Користувач	Тренування моделі	<p>— Система надає можливість користувачеві за натисканням кнопки «Тренування» виконати тренування моделі на обраному датасеті.</p>	Високий
Користувач	Введення даних для класифікації	<p>— Система надає можливість користувачеві обрати файл з дискового носія на комп'ютері.</p> <p>— Система надає можливість користувачеві ввести у текстове поле ім'я персонажа.</p>	Високий
Користувач	Визначення архетипу персонажа	<p>— Система надає можливість користувачеві натиснути кнопку «Визначити» та.</p> <p>— Система має зберігати виведений результат у базу даних.</p>	Високий



Таблиця 1.1 (закінчення)

Користувач	Перегляд результатів класифікації	<p>— Система надає можливість користувачеві отримати виведення даних про результати класифікації на екран.</p> <p>— Система надає можливість натиснути кнопку «Повернутись» для повернення до меню класифікації.</p>	Високий
Користувач	Перегляд інформації про архетипи	— Система надає можливість користувачеві натиснути елемент меню «Архетипи» для перегляду інформації про основні архетипи в літературі.	Низький
Користувач	Перегляд історії запусків класифікації	— Система надає можливість користувачеві переглянути інформацію з попередніх запусків системи.	Низький
Адміністратор	Перегляд та редагування історії запусків класифікації	— Система надає можливість адміністраторові змінювати записи у базі даних за допомогою таблиці на сторінці.	Високий

## 1.2 Огляд наявних аналогів

На сьогодні існує велика кількість різноманітних сервісів, що надають змогу користувачеві виконувати інтелектуальний аналіз тексту — процес автоматичного аналізу звичайних неструктурованих текстових документів комп'ютером з метою витягнення високоякісної структурованої інформації.

Більшість з них являє собою десктопні застосунки, або бібліотеки певних мов програмування. Розглянемо деякі з них.

GATE [3] (General Architecture for Text Engineering — основна архітектура для обробки тексту) — масштабний програмний продукт з відкритим кодом, який включає в себе інструменти для підтримки життєвого циклу ПЗ — від проектування і розробки до спільного використання великою кількістю серверів у цілях анотації документів. Також даний інструмент надає інтерфейс для застосунків всередині організації за допомогою бібліотеки об'єктів Java.

Серед переваг GATE не тільки розвинена екосистема, а також двадцятирічний досвід існування на ринку та переклад на десятки мов. З практичних переваг — обробка документів не перевантажує пам'ять, оскільки виконується послідовно, проте швидкість роботи системи при цьому значно падає.

Orange Інструмент для інтелектуального аналізу Orange [4] включає в себе розширення для роботи з неструктурованими масивами даних — в тому числі, з текстами. При цьому широкі можливості візуалізації Orange використовуються в цілях text mining. Подібні розширення зручні тим, що можна не відмовлятися від звичного ПЗ, коли з'являються нові задачі. Зручний графічний користувацький інтерфейс та інструменти візуального

					ДП 6306.00.000 ПЗ	Ажк.
Змн.	Арк.	№ докум.	Підпис	Дата		

програмування роблять Orange привабливим для користувачів, проте його «зав'язка» на Python може сподобатись не кожному.

Система інтелектуального аналізу даних KNIME [5] містить плагін для обробки тексту з назвою KNIME Text Processing. Плагін, як і уся система, має відкритий код і підтримує шестиступеневий процес обробки тексту — від читання до синтаксичного аналізу через розпізнавання сутностей, фільтрації та маніпуляції до підрахунку кількості слів, виділення ключових понять та візуалізації. Все це дає користувачеві широкі можливості роботи з текстом, проте він може знайти у KNIME пару вагомих недоліків. Наприклад, система не читає дані з MS Excel і не працює з технологією OLAP.

Розширення для роботи з текстами відомого інструменту RapidMiner [6] сподобається консерваторам. Воно відрізняється широкими можливостями у вирішенні завдань з text mining. Наголос робиться на статистичному аналізі, а дані вивантажуються з безлічі популярних форматів — в тому числі, PDF.

Безсумнівний плюс — графічний інтерфейс, що дозволяє управляти потоками даних, буквально просто «перетягуючи» їх з місця на місце. Правда, разом з перевагами, аддон успадкував і недоліки «батьківської системи» — обмежений обсяг даних, які можна обробити за допомогою безкоштовної версії ПЗ. Правда, для тексту один гігабайт — цифра чимала (згадати хоча б, що «Війна і мир» у форматі txt займає всього 300 КБ пам'яті), тож як мінімум в академічних колах цього недоліку можуть взагалі не помітити.

### 1.3 Постановка задачі

#### 1.3.1 Призначення розробки

Призначення розробки — визначення архетипу персонажа літературного твору.

### 1.3.2 Цілі та задачі розробки

Цілями розробки системи визначення архетипу персонажа літературного твору:

— спрощення процесу літературного аналізу творів.

Задачі розробки:

— тренування моделі на різних вибірках даних;

— визначення архетипу введеного персонажа у певному творі;

— визначення метрик результатів виконання машинного навчання на тестовій вибірці;

— отримання описової характеристики кожного архетипу;

— отримання історії використання програми;

### Висновок до розділу

На основі проведеного порівняння існуючих систем інтелектуального аналізу даних знайдено проблему відсутності аналогів розроблюваної системи. Було вивчено основні цілі класифікації тексту як напряму у сфері машинного навчання і визначено необхідність розробки поглибленого аналізу літературних творів як для розвитку сфери в цілому, так і для прикладного застосування системи.

					ДП 6306.00.000 ПЗ	Арх.
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

### 2.1 Вхідні дані

Вхідними даними даної системи є:

- дані, що надходять від користувача;
- дані, що містяться у базі, необхідні для тренування моделі.

Нижче наведено детальний розгляд даних.

Дані від користувача:

- обраний з системи користувача файл з розширенням \*.txt, який містить літературний твір;
- введенне користувачем в текстове поле ім'я персонажа твору.

Дані у базі, необхідні для тренування моделі:

- тексти літературних творів;
- імена та архетипи літературних героїв.

### 2.2 Вихідні дані

Вихідними даними є:

- архетип персонажа у літературному творі;
- описова характеристика наведеного архетипу.

### 2.3 Опис структури бази даних

На стадії проектування бази даних було виділено сутності «Архетип», «Текст», «Персонаж» та «Користувач». База даних складається з 4-х таблиць. Графічне зображення схеми бази даних знаходиться у графічних матеріалах.

Сутності схеми описані у таблицях 2.1-2.3

Сутність «Текст» заходиться у таблиці 2.1. Таблиця містить поля id, title, author, text — дані про текст для тренування моделі.

					ДП 6306.00.000 ПЗ	Аук.
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.1 — Опис таблиці «Text»

Код	Опис	Тип даних	Обов'язкове	Унікальне	Первинний ключ	Зовнішній ключ
id	Ідентифікатор тексту	integer	X	X	X	
title	Назва тексту	nvarchar(32)	X			
author	Автор тексту	nvarchar(32)	X			
text	Текст	nvarchar(MAX)	X			

Сутність «Архетип» заходиться у таблиці 2.2. Таблиця містить поля id, name, descr — дані про архетипи літературних персонажів для тренування та практичного використання моделі.

Таблиця 2.2 — Опис таблиці «Archetype»

Код	Опис	Тип даних	Обов'язкове	Унікальне	Первинний ключ	Зовнішній ключ

id	Ідентифікатор архетипу	integer	X	X	X	
----	------------------------	---------	---	---	---	--

Таблиця 2.2 (продовження)

name	Назва архетипу	nvarchar(32)	X			
descr	Опис архетипу	nvarchar(MAX)				

Сутність «Персонаж» заходиться у таблиці 2.3. Таблиця містить поля id, name, text\_id, arch\_id — дані про літературних персонажів для тренування та практичного використання моделі, а також зовнішні ключі, які вказують на архетип персонажа та текст, з якого персонаж походить.

Таблиця 2.3 — Опис таблиці «Character»

Код	Опис	Тип даних	Обов'язкове	Унікальне	Первинний ключ	Зовнішній ключ
id	Ідентифікатор персонажа	integer	X	X	X	
name	Ім'я персонажа	nvarchar(32)	X			
text_id	Зовнішній ключ тексту	integer	X			X

arch_id	Зовнішній ключ архетипу	integer				X
---------	-------------------------	---------	--	--	--	---

Сутність «Користувач» заходиться у таблиці 2.4. Таблиця містить поля id, username, password — дані користувача, необхідні для авторизації у системі.

Таблиця 2.4 — Опис таблиці «User»

Код	Опис	Тип даних	Обов'язкове	Унікальне	Первинний ключ	Зовнішній ключ
id	Ідентифікатор персонажа	integer	X	X	X	
username	Логін користувача	nvarchar(32)	X			
password	Пароль користувача	nvarchar(256)	X			

### Висновок до розділу

У даному розділі виконано планування та побудова структури бази даних, описано сутності у вигляді таблиць, наведено графічне зображення структури бази даних.



### 3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

#### 3.1 Змістовна постановка задачі

Класифікація даних [7] – загальна задача машинного навчання. Нехай деякі задані спостереження (об'єкти) належать до одного з двох класів. Завдання полягає в тому, щоб визначити, до якого класу будуть належати нові спостереження.

Аналогічно, класифікація тексту — це одна із задач інформаційного пошуку, яка полягає у відношенні документу до одної з декількох категорій на основі змісту документа.

Найчастіше класифікація текстів використовується для таких задач:

- перевірка на плагіат;
- вилучення спаму;
- визначення мови тексту;
- таргетування реклами;
- розподіл каталогів за жанрами.

На початку 2000-х років класифікація текстів була однією з часто обговорюваних тем, проте після винаходу підходу TF-IDF, який показує точність 95%-99.9%, ажіотаж стих. За такої точності слід більше звернути увагу саме на дані, які оброблюються, а не алгоритм.

У межах даного проекту розглядається особливий варіант класифікації за архетипами персонажів.

Користь даного виду класифікації не є такою ж очевидною, як, наприклад, класифікація електронних листів, каталогів, судових рішень тощо. Проте, класифікація літератури за персонажами є доволі розповсюдженою у вузьких спеціалізаціях. Одними з таких є:

- постановка театральних вистав;
- розробка кіносценаріїв;

					ДП 6306.00.000 ПЗ	Док.
Змн.	Арк.	№ докум.	Підпис	Дата		

- розробка описових характеристик сюжетів кінокартин, театральних постановок, телепередач;
- психологічні опитування.

При постановці кіно- та театральних сюжетів велику роль відіграють персонажі, тому для поглибленого розуміння мотивів та характерів персонажів буде актуальною розгляд їх психологічних архетипів.

Оскільки виділені архетипи засновані психологом і були створені саме для класифікації людських мотивів, тому і їх використання у психології є очевидним. На основі даних архетипів можна створювати психологічні тести, за результатами яких людина може сформувати певне уявлення про особливості власної психології, які раніше не були очевидними.

У даному проєкті на основі алгоритмів машинного навчання та класифікації текстів необхідно розробити модель, яка оброблює вхідні текстові дані і виконує безпосередньо класифікацію літературного твору з метою знаходження архетипу введеного користувачем персонажа.

### 3.2 Математична постановка задачі

Нехай  $X$  — простір об'єктів (наприклад,  $\mathbb{R}^n$ ),  $Y$  — кінцева множина імен класів (у даному випадку назви архетипів). Існує невідома цільова залежність — відображення  $y^*: X \rightarrow Y$ , значення якої відомі лише на об'єктах кінцевої тренувальної вибірки  $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$ . Необхідно побудувати алгоритм:  $a: X \rightarrow Y$ , здатний класифікувати довільний об'єкт  $x \in X$ .

### 3.3 Обґрунтування методу розв'язання

Метод опорних векторів (англ. Support Vector Machine, SVM) [8] - один з найбільш популярних методів навчання, який застосовується для розв'язання задач класифікації і регресії. Основна ідея методу полягає в знаходженні гіперплощини в  $N$ -мірному просторі ( $N$  — кількість ознак), що розділяє

об'єкти вибірки оптимальним способом. Алгоритм працює в припущенні, що чим більше відстань між розділяючою гіперплощиною і об'єктами поділюваних класів, тим меншою буде середня похибка класифікатора.

Гіперплощини – це такі площини, що допомагають класифікувати точки даних. Точки даних, що падають з будь-якої сторони гіперплощини, можна віднести до різних класів. Також розмірність гіперплощини залежить від кількості ознак. Якщо кількість вхідних функцій дорівнює 2, то гіперплощина — це всього лінія. Якщо кількість вхідних ознак дорівнює 3, то гіперплощина стає двовимірною площиною. Візуалізація гіперплощин розмірністю 3 і більше є набагато складнішою. Графічне зображення гіперплощин розмірності 1 та 2 зображено на рисунку 3.1.

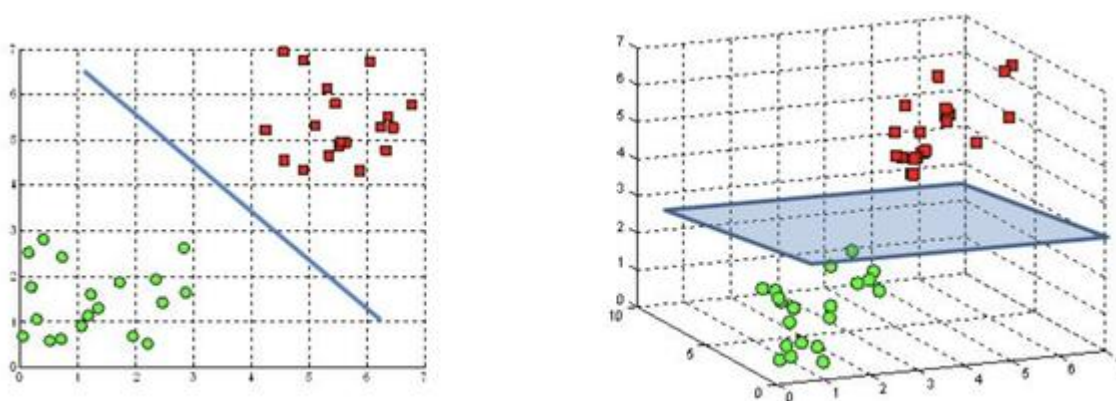


Рисунок 3.1 — Гіперплощини з розмірністю 1 та 2 відповідно

Опорні вектори — це точки даних, які знаходяться ближче до гіперплощини і впливають на її положення та орієнтацію [9].. Використовуючи ці вектори, максимально збільшується запас класифікатора. Видалення опорних векторів змінить положення гіперплощини. Поняття опорних векторів та гіперплощин є основними при розробці алгоритму методом опорних векторів.

Одна з переваг SVM-методу в тому, що так як він намагається визначити оптимальний напрямок поділу простору ознак, розглядаючи комбінації ознак, він досить стійкий до великих розмірностей. Текстові дані ідеально підходять для класифікатора SVM через розрідженість даних великої розмірності, що є відмінною рисою текстів.

Недоліки методу: складна інтерпретованість параметрів алгоритму і нестійкість по відношенню до «викидів» у вихідних даних.

### 3.4 Опис методів розв'язання

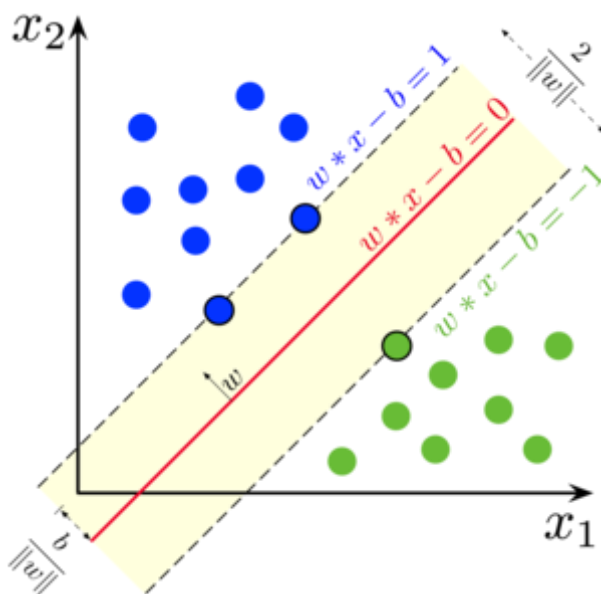
Оберемо пряму, максимально далеко проходить від точок. Відстань від неї до найближчої точки з кожного боку максимальна.

Якщо така пряма існує, то її називають гіперплощиною максимальної різниці, і лінійний класифікатор, що її визначає, відповідно класифікатор максимальної різниці або перцептрон оптимальної стабільності (стійкості).

Опорні вектори — це точки, для яких відстань до гіперплощини  $\frac{1}{\|w\|}$ .

Вони є ефективними елементами на навчальній вибірці.

Графічне зображення опорних векторів представлено на рисунку 3.2.



### Рисунок 3.2 — Візуалізація опорних векторів

Важливо зазначити, що якщо всі точки, крім опорних векторів, видалити, алгоритм методу опорних векторів залишиться колишнім. Ця властивість робить SVM унікальним, несхожим на всі інші методи, такі як kNN, LLSF, NNet і NB, де всі точки тренованої вибірки використовуються для оптимізації функції. Теоретична відмінність призводить до значної розбіжності між SVM та іншими методами на практиці.

Цей підхід узагальнюється на багатовимірний випадок.

У даному проекті використовується лінійний метод опорних векторів для виконання класифікації, проте слід описати і нелінійний варіант для розуміння різниці між ними у загальному вигляді.

#### Лінійний метод опорних векторів

Дано спостереження для навчання  $D$ , набір складається з  $n$  об'єктів:

$$D = \{(x_i, y_i) | x_i \in R^p, y_i \in \{-1, 1\}\}_{i=1}^n \quad (3.1)$$

де  $y_i$  приймає значення -1 або 1, визначаючи, до якого класу належить кожна точка  $x_i$ . Кожна точка  $x_i$  — вектор розмірності  $p$ .

Необхідно знайти гіперплощину максимальної різниці, яка розділяє спостереження, що мають  $y_i = 1$  від об'єктів  $y_i = -1$ .

Будь-яку гіперплощину можна записати як безліч точок  $x$ , що задовольняють:

$$w * x - b = 0 \quad (3.2)$$

де  $*$  - скалярний добуток нормалі до гіперплощини на вектор  $x$ .

Параметр  $\frac{b}{||w||}$  визначає зміщення гіперплощини від початку координат вздовж нормалі  $w$ .

Якщо навчальні дані є лінійно нероздільні, можна вибрати дві паралельні гіперплощини таким чином, що вони розділять безліч точок на 2 класи, і точок між ними не буде.

Потім намагаються максимізувати дистанцію між ними, одночасно роблячи поворот і паралельно зрушення паралельних прямих.

Область, обмежена 2 гіперплощинами, називається різницею (маржею).

Ці гіперплощини можуть бути описані рівняннями:

$$w * x - b = 1 \quad (3.3)$$

$$w * x - b = -1 \quad (3.4)$$

Знаходимо відстань між гіперплощинами:  $-\frac{2}{||w||}$ .

Для того, щоб дистанція була максимальною, мінімізуємо  $||w||$ .

Щоб виключити всі точки зі смуги, ми повинні перекоонатися, що для всіх спостережень справедливим є:

$$w * x_i - b \geq 1, \text{ для } x_i \text{ з першого класу} \quad (3.5)$$

$$w * x_i - b \leq -1, \text{ для } x_i \text{ з другого класу} \quad (3.6)$$

Еквівалентно для  $0 \leq i \leq n$ :

$$y_i(w * x_i - b) \geq 1 \quad (3.7)$$

Далі вирішується задача оптимізації:

$$||w|| \rightarrow \min \quad (3.8)$$

$$y_i(w * x_i - b) \geq 1, \text{ для } 0 \leq i \leq n \quad (3.9)$$

### Нелінійний метод

Для створення нелінійного класифікатора використовується довільна функція ядра. Кожний скалярний добуток замінюється на нелінійну функцію ядра. Це дозволяє знаходити гіперплощину максимальної різниці в трансформованому просторі функцій.

Зміна може бути нелінійною і трансформуватися в простір з більш високою розмірністю.

Незважаючи на те, що класифікатор є гіперплощиною в багатовимірному просторі функцій, він може бути нелінійним в вихідному просторі навчальної вибірки.

### Багатокласовий метод опорних векторів

Найбільш популярний підхід — перехід від завдання класифікації на множину класів до множинної задачі розбиття на два класи. Даний метод використовує 2 стратегії:

- один проти всіх;
- один проти одного.

У першому випадку навчається  $N$  класифікаторів, де  $N$  — кількість класів. Класифікатор з найвищим значення функції виходу привласнює новий об'єкт до певного класу.

У випадку «один проти одного» також навчається  $N$  класифікаторів, тільки в цьому разі об'єкт присвоюється до того класу, до якого його віднесло більшість класифікаторів.

### Висновок до розділу

У даному розділі було описано змістовну та математичну постановки, наведено алгоритм розв'язання поставленої задачі.

## 4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 4.1 Засоби розробки

Розробка системи проведена на базі технології Python [10]. Дана мова програмування є високорівневою і добре підходить для виконання задач, пов'язаних безпосередньо з веб-застосуваннями та системами машинного навчання й штучного інтелекту. Технологія не дає такого ж виграшу у швидкості роботи, як такі низькорівневі мови, як С або С++. Проте розробникові надається змога зосередитись на алгоритмах програми, не витрачаючи велику кількість часу на написання коду.

Найбільша перевага технології Python — це її зовнішні бібліотеки. Вони значно розширюють можливості мови програмування і є невід'ємною частиною Python. Одними з таких є Flask [11] та scikit-learn [12].

Flask — веб-фреймворк на базі Python. Основною його перевагою є його простота розробки застосувань та взаємодія з клієнтською частиною. Інтерфейс даного проекту повністю розроблено за допомогою вищевказаного фреймворка.

Scikit-learn — бібліотека, що надає корисні інструменти для розробки моделей машинного навчання. У даній системі необхідно якісно оброблювати текст для правильної роботи моделі та отримання достовірних результатів класифікації, тому sklearn є одним із допоміжних методів, використаних при розробці системи.

Локальний сервер, використаний для розробки, є на базі операційної системи Fedora 31, що є однією з розширень Unix-систем. Операційні системи на базі Linux широко використовуються розробниками у світі через свою низькорівневість. Контроль над усією системою може відбуватись за допомогою терміналу, тобто такі операції як запуск сервера, робота з



системою управління базою даних та системою контролю версій здійснюються в одному застосуванні.

Веб-сервер розгорнуто на базі операційної системи Ubuntu 16.04 за допомогою онлайн-сервісу DigitalOcean. Недоліком даного сервісу є те, що він не безкоштовний, на відміну від таких аналогів як Heroku.

Створення діаграм виконано за допомогою онлайн-сервісу draw.io. Дана платформа не вимагає скачування десктоп-застосування, має зручний та зрозумілий інтерфейс, а також паттерни більшості UML-діаграм, необхідних для розробки програмного продукту.

У якості системи контролю версій обрано Git, платформа — GitHub. Використання системи контролю версій є загально прийнятим підходом до розробки програмного забезпечення через можливість зберігати прогрес виконання роботи, незалежно від стану машини, на якій проводилась розробка. Також розгортання веб-застосування вимагає використання Git для зручного оновлення функціоналу системи.

## 4.2 Вимоги до технічного забезпечення

### 4.2.1 Загальні вимоги

Для правильної роботи системи класифікації та тренування моделі необхідний сервер з наступними мінімальними параметрами:

- процесор з частотою 1,6 ГГц або вище;
- об'єм оперативної пам'яті 2 ГБ або більше;
- жорсткий диск об'ємом не менше 20 ГБ;
- операційна система Linux (Fedora 31, Ubuntu 16.04 або 18.04).

Для користування системою на стороні клієнта мають бути виконані наступні вимоги:

— наявність веб-браузера Google Chrome, Firefox, Microsoft Edge, Opera.

### 4.3 Архітектура програмного забезпечення

Система класифікації архетипу персонажа літературного твору є веб-застосуванням з використанням алгоритмів машинного навчання, тому для її розробки використано мову програмування Python.

Класифікація архетипу персонажа відбувається у декілька етапів. Перший етап — збір тренувальних датасетів для моделі. Оскільки для тренування системи розпізнавати архетип персонажа необхідно спочатку вказати в тренувальних даних, які бувають архетипи та навести приклади, усі датасети зібрано «вручну» шляхом аналізу літературних творів та вказуванням архетипів основних дійових осіб. Кожен датасет містить список з імен героїв та їх архетипів на початку файла. Вигляд датасетів у системі зображено на рисунку 4.1.

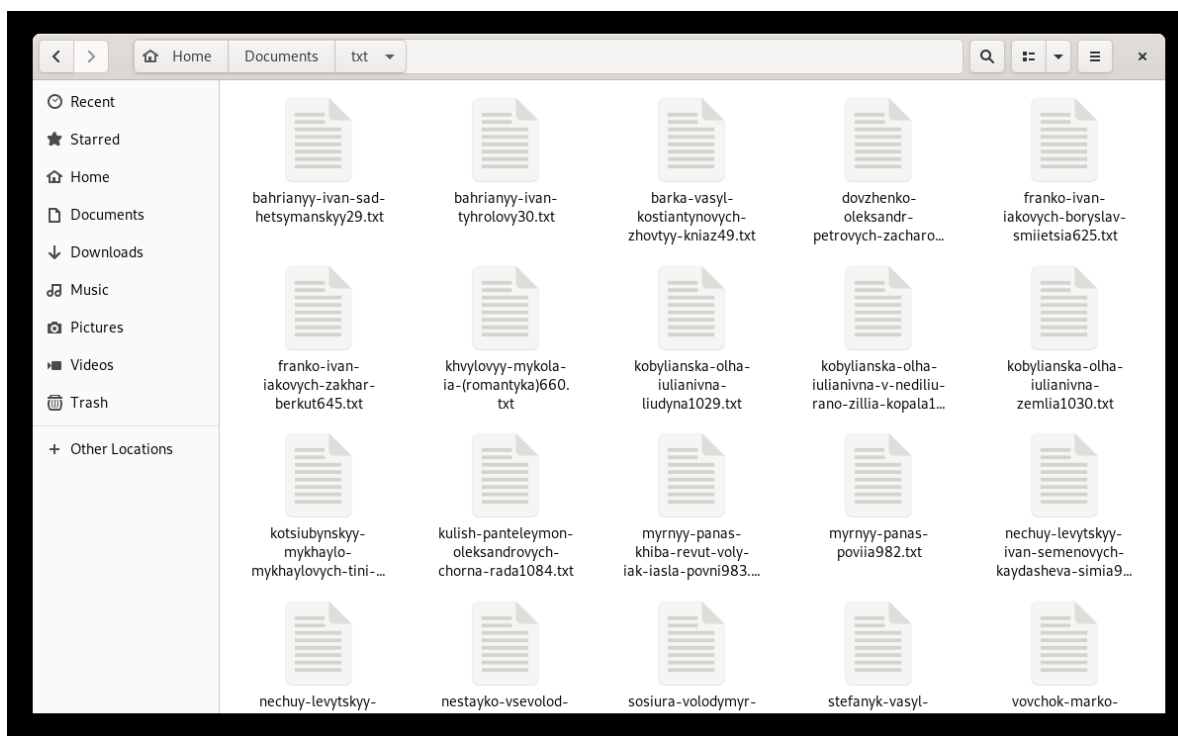


Рисунок 4.1 — Датасет української прози

Приклад одного файлу із датасету зображено на рисунку 4.2.

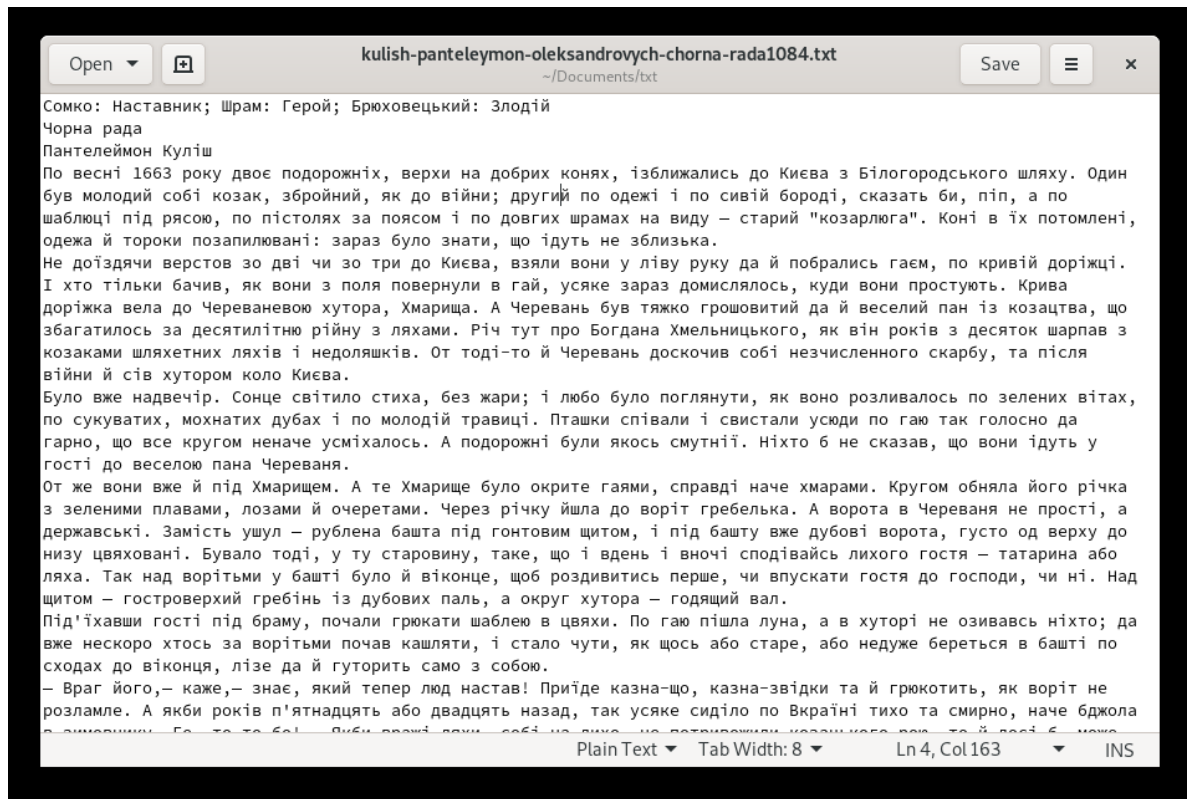


Рисунок 4.2 — Пантелеймон Куліш — Чорна Рада

Для тренування моделей було обрано датасети з української та зарубіжної літератури. Як для української, так і для зарубіжної було обрано більше 50 творів по 2-5 персонажів у кожному. Використано роботи 22-х авторів української літератури та 12-х зарубіжної.

Після підготування датасету програмі необхідно його обробити для виконання класифікації. Спочатку виконується обробка тексту на знаходження абзаців з участю вказаних персонажів, потім формуються нові тексти окремо для кожного персонажа і саме вони беруть участь у тренуванні моделі на встановлення архетипів.

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Відфільтровані тексти зчитуються програмою і відбувається перетворення тексту — векторизація. Вона складається з п'яти етапів:

- токенізація — розбиття довгих ділянок тексту на більш дрібні (абзаци, пропозиції, слова);
- нормалізація — приведення тексту до «очищеного» вигляду (єдиний реєстр слів, відсутність знаків пунктуації, розшифровані скорочення, словесне написання чисел і т.д.). Це необхідно для застосування уніфікованих методів обробки тексту;
- стеммізація — приведення слова до його кореня шляхом усунення суфіксів, приставок, закінчень тощо;
- лематизація — приведення слова до смислової канонічної форми слова (інфінітив для дієслова, називний відмінок однини - для іменників і прикметників);
- чистка — видалення стоп-слів, які не несуть смислового навантаження (артиклі, займенники, сполучники, прийменники і т.д.).

Хоча мова програмування Python надає готові рішення для даної задачі, скористатись ними неможливо через використання української мови у датасетах та розробці. Готові алгоритми не розпізнають кодування і методи не виконуються як слід. Тому алгоритми для виконання вищевказаних дій розроблено власноруч саме для української мови.

Наступним кроком є застосування алгоритму машинного навчання, опис якого наведено у розділі 3.

Для взаємодії з користувачем обрано архітектуру системи на основі веб-застосунку.

Веб-застосунок — це клієнт-серверне рішення, у якому клієнт виконує взаємодію з сервером за допомогою веб-браузера. Основна частина роботи системи полягає саме у серверній частині. Перевагою для користувачів є те,

що робота системи не залежить від машини клієнта, необхідна лише наявність одного з вищевказаних веб-браузерів для коректного відображення інтерфейсу і можливості використання усіх запропонованих функцій. Розробка не вимагає налаштування системи для роботи під різними операційними системами, наприклад, Windows, Linux, MacOS.

Розробка системи відбувається на локальній машині, після чого за допомогою системи контролю версій Git завантажується на веб-сервер та має змогу отримувати оновлення, уникаючи безпосередньої взаємодії з ОС веб-сервера.

Схема розгортання веб-застосунку показана на рисунку 4.1.

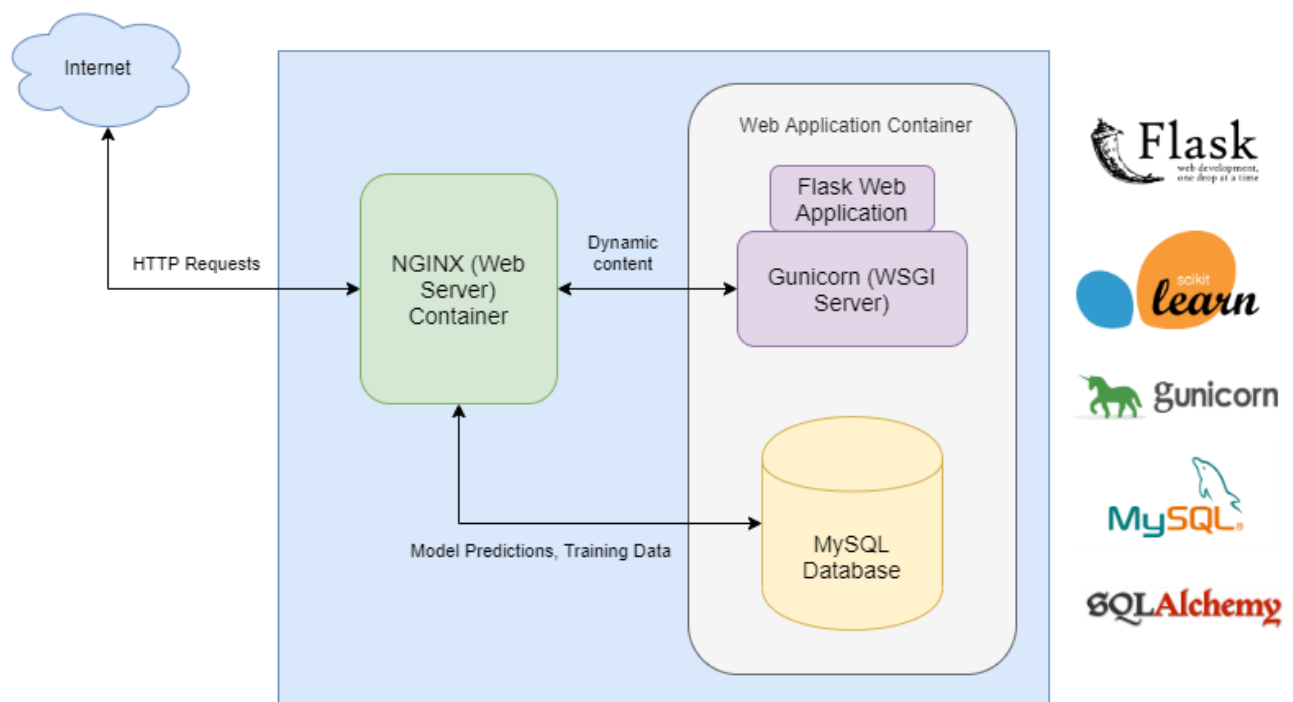


Рисунок 4.3 — Схема веб-розгортання системи

Після створення початкової версії системи, розробник завантажує її до репозиторію GitHub [13]. Наступним кроком є створення веб-сервера у системі DigitalOcean [14] та його налаштування: встановлення NGINX [15], який надає

змогу системі працювати без вмикання її «вручну», а також надалі буде взаємодіяти з локальною машиною за допомогою gunicorn — Python WSGI HTTP сервера. Далі на сервер завантажується проект з репозиторію GitHub та виконується міграція бази даних з локальної машини.

### 4.3.1 Діаграма класів

В процесі розробки системи класифікації архетипу персонажа літературного твору використовувався як об'єктно-орієнтований підхід, так і функціональна парадигма програмування.

Об'єктно-орієнтоване програмування використовується для взаємодії системи з базою даних шляхом створення моделі, яка є аналогом представлення даних у базі MySQL. Кожна таблиця у базі даних відповідає класу у моделі.

Функціональне програмування надає змогу розроблювати веб-застосунки на базі мови програмування Python. Кожна окрема сторінка має функцію з декоратором-обробником, які реалізують функціонал застосунку, що є загальноприйнятим підходом при створенні систем за допомогою даної технології.

Для розробки програмного продукту було створено класи, які наведені у таблиці 4.1.

Таблиця 4.1 – Опис класів

Клас	Опис
User	Клас, що описує користувача. Необхідний для функції авторизації.
Book	Клас, що описує літературний твір.
Archetype	Клас, що описує архетипи персонажів.

Клас	Опис
Person	Клас, що описує персонажа, для якого проводиться класифікація.

Опис атрибутів наведених класів зазначено у таблиці 4.2.

Таблиця 4.2 – Опис атрибутів класів

Клас	Атрибут	Тип даних	Опис
User	id	int	Ідентифікатор запису
	username	string	Логін користувача
	password	string	Пароль користувача
Archetype	id	int	Ідентифікатор запису
	name	string	Назва архетипу
	person	relationship	Особливий тип даних для зв'язку між таблицями (аналог зовнішніх ключів)

Book	id	int	Ідентифікатор запису
	name	string	Назва твору
	author	string	Автор твору

Таблиця 4.2 (продовження)

	person	relationship	Особливий тип даних для зв'язку між таблицями (аналог зовнішніх ключів)
Person	id	int	Ідентифікатор запису
	name	string	Ім'я персонажа
	book_id	int	Зовнішній ключ до класу Book
	archetype_id	int	Зовнішній ключ до класу Archetype

Графічне зображення діаграми наведено у графічних матеріалах.

#### 4.3.2 Діаграма послідовності

Для побудови діаграми послідовності необхідно розглянути послідовність дій користувача при взаємодії з системою.



Першим кроком є вхід користувача в систему та початок тренування моделі на підготованих датасетах. Якщо користувач не проводить тренування, використовується датасет за замовчуванням — українська проза. Це не є критичним, проте напряду впливає на точність класифікації моделі, адже якщо тренування проведене на даних з української літератури, почерк зарубіжних авторів може бути неправильно «зрозумілий» системою.

У разі правильної послідовності дій, користувач запускає тренування системи, запускається відповідний алгоритм, описаний вище, і система навчається класифікувати текст за наявними даними.

Наступним кроком є безпосередньо класифікація архетипу персонажа. Користувачеві необхідно додати файл з дискового носія комп'ютера з текстом твору з розширенням \*.txt, а також вказати ім'я персонажа, для якого визначатиметься архетип. У разі невірної формату файлу або незаповнених полів система повідомить користувача про помилку і запропонує ввести дані ще раз. У разі введення коректних даних система виконає класифікацію і виконає перехід користувача на сторінку результатів роботи.

У результатах роботи вказується назва файлу, ім'я героя, його архетип, а також точність класифікації. Дані записуються до бази і можуть бути переглянуті в меню «Історія класифікацій».

Система має сторінки, які не є обов'язковими, проте надають корисну інформацію. Це сторінки «Архетипи» та «Історія». При переході на них користувачеві надається інформація про існуючі архетипи в літературі та історія класифікацій системи відповідно.

Для адміністратора надається можливість входу в систему для можливості редагування бази даних шляхом взаємодії з інтерфейсом. При натисненні на елемент меню «Увійти» користувач переходить на сторінку авторизації, де йому надається змога ввести логін та пароль, щоб увійти у

систему. Для авторизованих користувачів кнопка «Увійти» змінюється на «Вийти».

Згідно з описаними діями користувача побудовано UML-діаграму послідовності, яка представлена у графічних матеріалах.

Графічне зображення діаграми наведено у графічних матеріалах.

#### 4.3.3 Діаграма компонентів

Система класифікації архетипу персонажа літературного твору поділяється на такі компоненти:

- Сервер MySQL — зберігання даних про архетипи, твори та результати класифікацій, проведених користувачами, а також авторизаційні дані користувачів;
- Parser — містить набір пакетів для перетворення датасету до цільової форми, що містить лише дані про потрібного персонажа;
- Classifier — містить набір пакетів для виконання математичної складової проекту;
- Flask Application — містить набір пакетів для реалізації серверної частини застосунку;
- Web-server — містить набір пакетів для розгортання веб-застосунку у мережі;
- UI — містить набір пакетів, що становлять клієнтську частину веб-застосунку.

Необхідно розглянути кожен компонент і встановити залежності між ними.

Графічне зображення діаграми наведено у графічних матеріалах.

Сервер MySQL відповідає за зберігання даних про такі сутності, як «Архетип», «Літературний твір», «Персонаж» та «Користувач». Містить усі необхідні функції, що застосовуються для проведення аналізу щодо

					ДП 6306.00.000 ПЗ	Адк.
Змн.	Арк.	№ докум.	Підпис	Дата		

попереднього використання програми і створення статистичних висновків про роботу системи. Зв'язок з програмним кодом відбувається за допомогою пакета SQLAlchemy, який є зовнішньою бібліотекою мови програмування Python.

Компонент Parser відповідає за обробку наданого системою тексту, незалежно від виконаної операції — тренування чи класифікація. Під обробкою мається на увазі виділення з тексту окремих персонажів, які зазначені користувачем для відкидання зайвих розділів твору і створення готового до класифікації тексту.

Компонент Classifier відповідає за математичне забезпечення програмного коду, тобто виконує усі необхідні дії для отримання результату класифікації.

В першу чергу виконується векторизація тексту — токенізація, нормалізація, стеммізація, лематизація та чистка. Наступним кроком є використання алгоритмів машинного навчання на обробленому тексті для виконання задачі класифікації — передбачення належності наданого персонажа до однієї з категорій, заданих у процесі тренування моделі.

Компонент Flask Application відповідає за серверну частину застосунку, тобто є «мостом» між математичної складовою програмного продукту та його веб-стороною. Проводить такі операції як авторизація користувача, перехід між сторінками, а також надає можливість шляхом взаємодії з інтерфейсом надавати можливість здійснювати тренування та класифікацію.

Компонент Web-Server відповідає за розгортання веб-застосунку та безперервну роботи системи у мережі. У якості веб-сервера обрано nginx, який зарекомендував себе як надійний HTTP-сервер і зворотній проксі-сервер. Основною його функціональністю є обслуговування статичних запитів, розподіл навантаження та відмовостійкість, модульність та підтримка uwsgi.

					ДП 6306.00.000 ПЗ	42к.
Змн.	Арк.	№ докум.	Підпис	Дата		

Компонент UI відповідає за взаємодію користувача із системою. Створено набір пакетів, кожен з яких надає певний функціонал на стороні клієнта.

#### 4.3.4 Специфікація функцій

Оскільки функціонал веб-застосунку розроблено в основному з використанням функціонального підходу, наявна велика кількість методів, що виконують дії над веб-інтерфейсом застосунку, а також алгоритмічною частиною. Специфікація функцій наведена у таблиці 4.3.

Таблиця 4.3 – Специфікація функцій

Назва функції	Аргументи	Опис аргументів	Опис функції
@app.route('/')	/	Сторінки у системі	Декоратор функцій, що здійснює взаємодію коду з серверної частини з клієнтською
logout()	—	—	Виконує вихід користувача з системи, переносить його на головну сторінку
load_user(user_id)	user_id	Ідентифікатор користувача	Завантажує користувача до системи
login()	—	—	Передхід користувача на сторінку авторизації

Таблиця 4.3 (продовження)

archetypes()	—	—	Перехід користувача на сторінку «Архетипи»
edit_book(id)	id	Ідентифікатор літературного твору	Зміна даних у таблиці та базі даних
delete_person(id)	id	Ідентифікатор персонажа	Видалення запису з таблиці та бази даних
history()	—	—	Перехід на сторінку «Історія»
hash_password(password)	password	Текстовий рядок — пароль користувача	Шифрування паролю для зберігання у базі даних
verify_password(stored_password, provided_password)	stored_password	Пароль з бази даних	Перевірка правильного введення паролю
	provided_password	Пароль, введений користувачем	
preprocess(dataset)	dataset	Текст літературного твору	Виділення у літературному творі персонажів та формування нових текстів

Таблиця 4.3 (закінчення)

load_files(path, categories, encoding)	path	Шлях до текстового файлу у системі	Завантаження файлів та їх переконвертування в об'єкт
	categories	Категорії класифікації	
	encoding	Кодування	
svm()	—	—	Виконання алгоритму опорних векторів
fit(data, target)	data	Переконвертовані дані у вигляді об'єкта	Знаходження паттернів у даних
	target	Об'єкт-ціль, очікуваний результат класифікації	
predict()	—	—	Виконання передбачення, повертає те значення, яке бачить користувач в результаті класифікації
vectorizer()	—	—	Виконання попередньої обробки тексту

### Висновок до розділу

У даному розділі описана архітектура системи, вказано засоби розробки, визначено вимоги до технічного забезпечення сервера та клієнта, описано архітектуру системи.

Наведено детальний опис UML-діаграм класів, послідовності та компонентів, а також класів та функцій програмного продукту.

## 5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

### 5.1 Керівництво користувача

У даному розділі наведено основні дії користувача у системі, результати виконаних дій, ознайомлення з інтерфейсом користувача за допомогою скріншотів виконання програми та текстового опису наявних функцій системи.

Метою користування є отримання класифікації архетипу персонажа певного літературного твору, обраного самим користувачем, а також розширення знань про архетипи та їх практичного використання.

#### 5.1.1 Головне меню

Головне меню, скріншот якого зображено на рисунку 5.1 представляє собою сторінку з коротко викладеною інформацією про функціонал системи. Згору розташоване фіксоване меню навігації по системі, яке включає в себе такі сторінки як «Тренування», «Класифікація», «Архетипи», «Історія» та «Увійти». Також користувач має можливість одразу перейти до тренування системи, скориставшись кнопкою «Почати роботу», що знаходиться в основній частині меню.

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

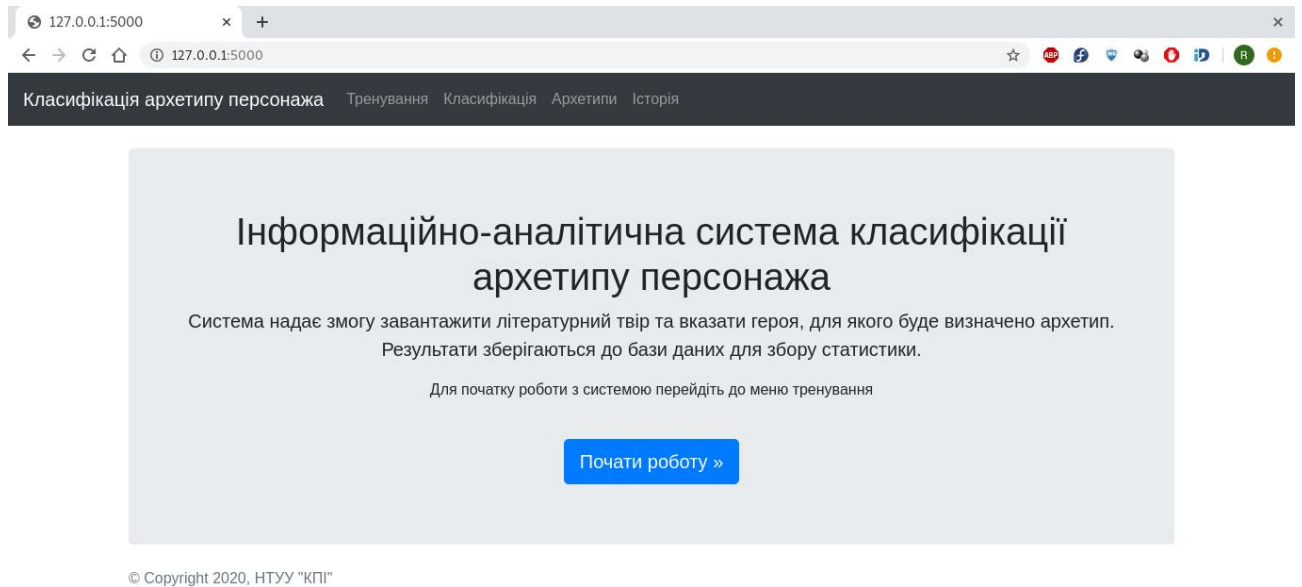


Рисунок 5.1 — Головна сторінка програми

Кнопка меню «Увійти» призначена адміністраторові для авторизації у системі. Звичайному користувачеві не потрібно бути авторизованим для використання системи.

### 5.1.2 Авторизація

Після натиснення кнопки «Увійти» відкривається сторінка авторизації, яка зображена на рисунку 5.2.

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		



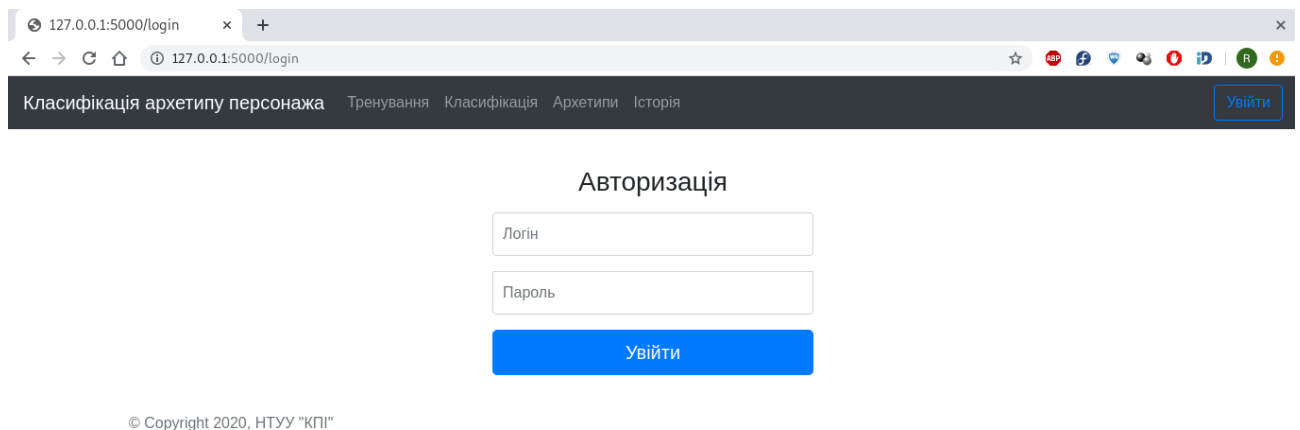


Рисунок 5.2 — Сторінка авторизації

На даній сторінці розташовано такі елементи, як два текстових поля для введення логіна та пароля користувача, а також кнопка увійти, натискання якої перевіряє логін та пароль та авторизує адміністратора у системі.

Для початку роботи з системою необхідно спочатку натиснути кнопку «Почати роботу» у головному меню, або скористатись меню вгорі і обрати елемент «Тренування». Це зумовлено тим, що задача класифікації потребує попереднього тренування моделі для виконання поставленої мети.

### 5.1.3 Тренування

Після натиснення елементу меню «Тренування» здійснюється перехід на сторінку, яка зображена на рисунку 5.3.

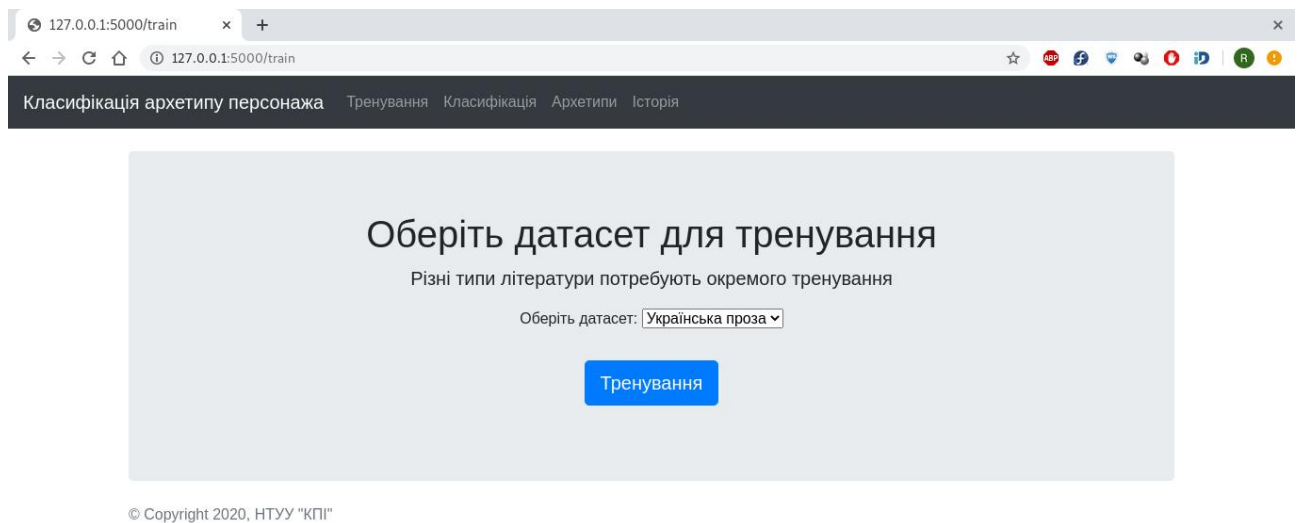


Рисунок 5.3 — Сторінка тренування системи

На даній сторінці користувач має можливість обрати один з датасетів, наявних у системі для підвищення точності класифікації, в залежності від очікуваного результату. Сторінка містить випадające меню з датасетами, які включають українську та зарубіжну прозу, а також кнопку «Тренування», яка безпосередньо виконає функцію тренування моделі у системі.

Після виконання тренування моделі, користувачеві необхідно перейти до меню класифікація, скориставшись меню згору та обравши елемент «Класифікація».

#### 5.1.4 Класифікація

Після натиснення кнопки «Класифікація» відбувається перехід до меню класифікації, що зображене на рисунках 5.4 та 5.5. На сторінці розміщені такі кнопки, як «Обрати файл» та «Визначити», а також текстове поле «Ім'я» для введення імені персонажа, для якого проводиться класифікація.

					ДП 6306.00.000 ПЗ	49к.
Змн.	Арк.	№ докум.	Підпис	Дата		

Класифікація архетипу персонажа Тренування Класифікація Архетипи Історія

## Завантажте файл для класифікації

Оберіть \*.txt файл з текстом українською мовою та введіть ім'я персонажа

Choose File dovzhenko-ol...-ohni876.txt

Ім'я Лаврін

Визначити

© Copyright 2020, НТУУ "КПІ"

Рисунок 5.4 — Сторінка класифікації

Класифікація архетипу персонажа Тренування Класифікація Архетипи Історія Вийти

## Завантажте файл для класифікації

Оберіть \*.txt файл з текстом українською мовою та введіть ім'я персонажа

Choose File Harri-Potter-i...lin-Roling.txt

Ім'я Гаррі

Визначити

© Copyright 2020, НТУУ "КПІ"

Рисунок 5.5 — Сторінка класифікації

Користувач має можливість завантажити файл — текстовий файл, що містить літературний твір — з власного дискового носія для проведення

					ДП 6306.00.000 ПЗ	50к.
Змн.	Арк.	№ докум.	Підпис	Дата		

класифікації на його основі. У текстовому поясненні вказано, що файл мусить мати розширення \*.txt, інакше виконання класифікації не відбуватиметься. Також необхідно вказати ім'я героя, для якого визначатиметься архетип в обраному літературному творі. Після заповнення всіх полів необхідно натиснути кнопку «Виконати» для здійснення класифікації. Дана дія завантажує сторінку результатів класифікації.

### 5.1.5 Результати класифікації

Результати класифікації зображені на рисунках 5.5 та 5.6 для української та зарубіжної літератури відповідно. Сторінка результатів має текстовий опис виконаної класифікації архетипу, а саме файл, з якого узято текст літературного твору, ім'я класифікованого персонажа, його визначений архетип, точність класифікації і кнопка «Повернутись».

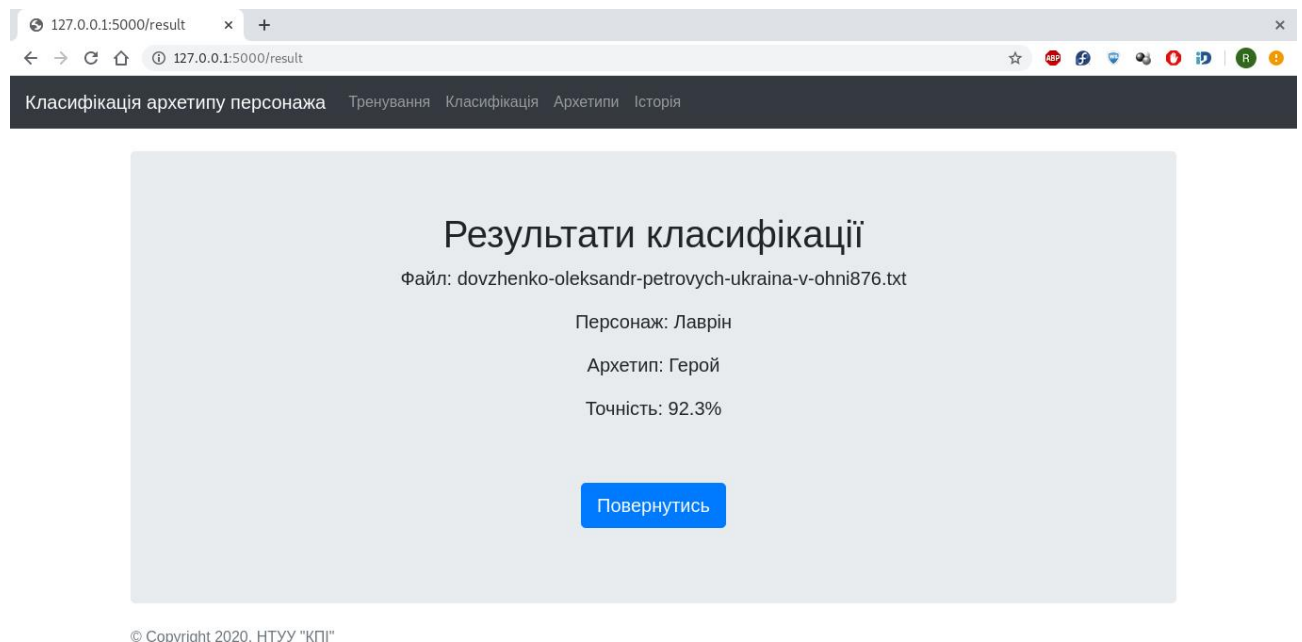


Рисунок 5.6 — Сторінка результатів класифікації для української прози

					ДП 6306.00.000 ПЗ	Адк.
Змн.	Арк.	№ докум.	Підпис	Дата		

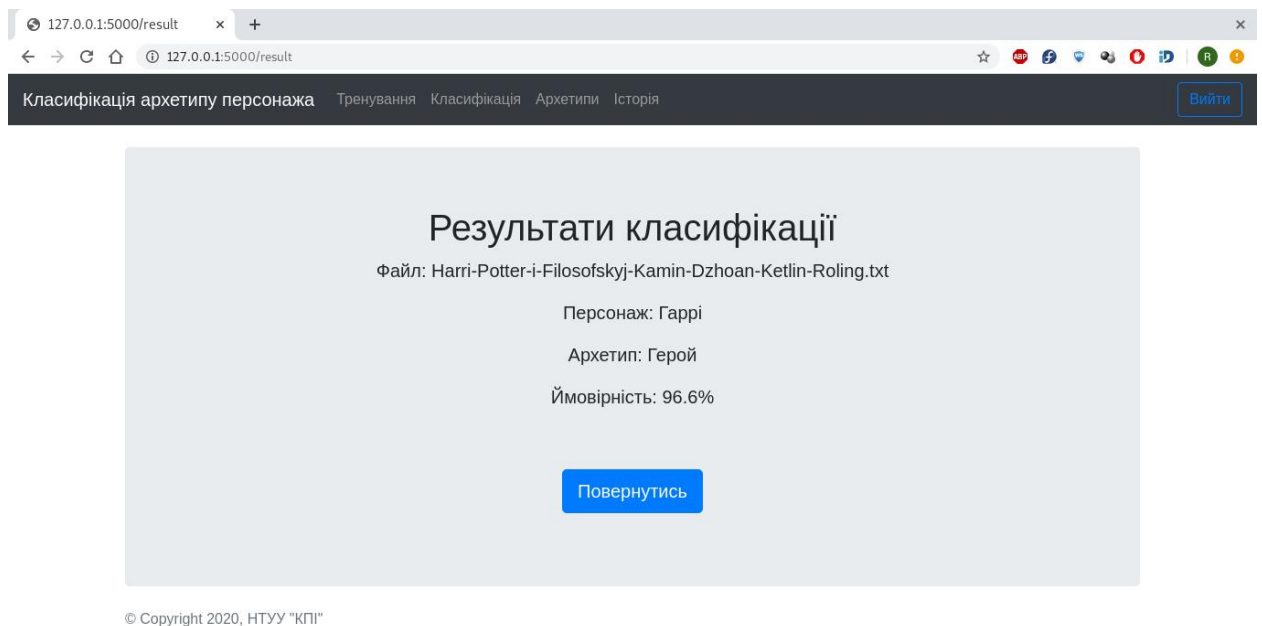


Рисунок 5.7 — Сторінка результатів класифікації для української прози

На скриншоті бачимо виконану класифікацію архетипу відповідно до даних, введених у систему, що зображені на рисунках 5.4 та 5.5 відповідно. При натисненні кнопки «Повернутись» завантажується сторінка «Класифікація», що описана вище.

### 5.1.6 Архетипи

Користувач має можливість обрати елемент меню «Архетипи» для перегляду короткого опису кожного з основних архетипів у літературі. Сторінка «Архетипи» зображена на рисунках 5.8 та 5.9. Дана сторінка не має активних елементів, таких як кнопки, вона є описовою, створена з метою надати базову характеристику використаних у системі архетипів.

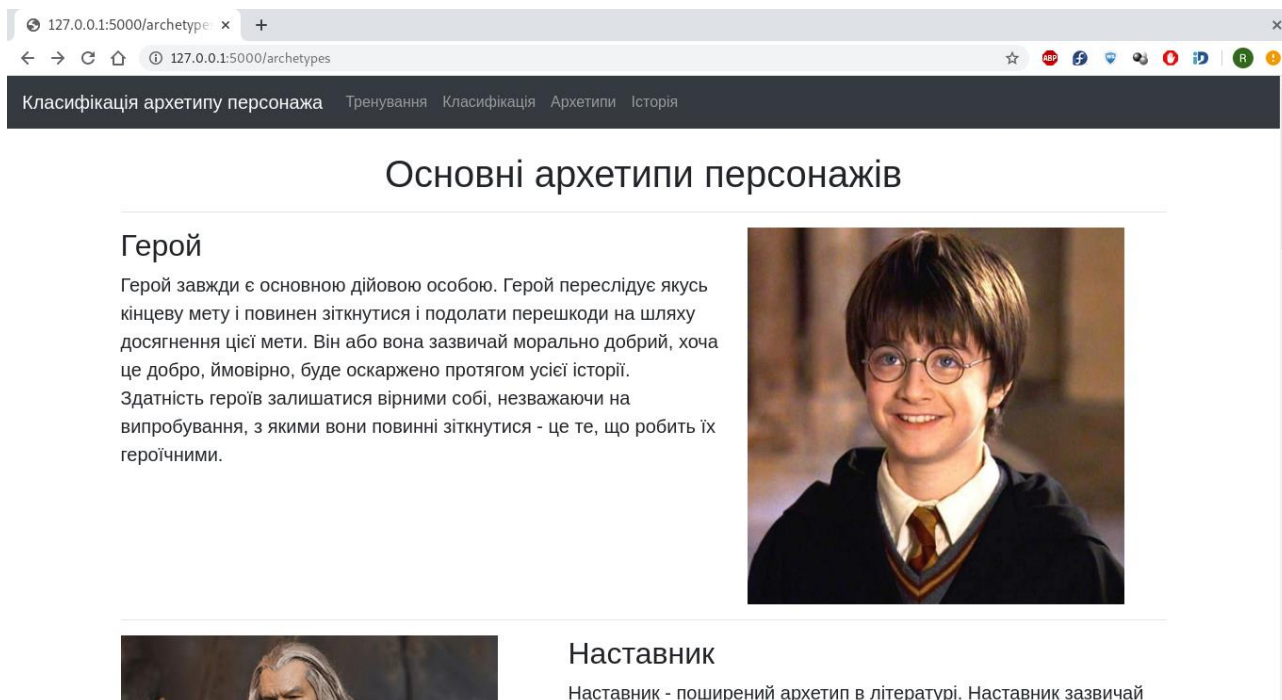


Рисунок 5.8 — Сторінка описової характеристики архетипів (1 частина)

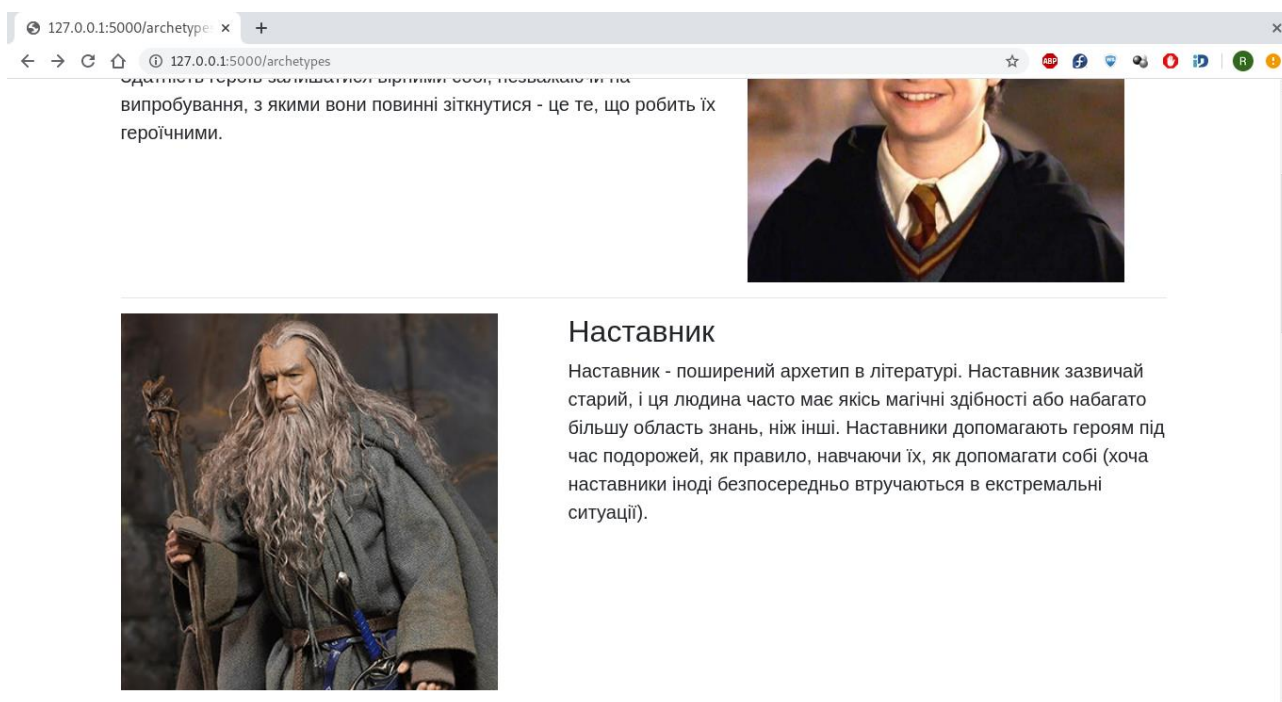


Рисунок 5.9 — Сторінка описової характеристики архетипів (2 частина)

### 5.1.7 Історія класифікацій

Користувач має можливість обрати елемент меню «Історія» для перегляду історії використання системи. Сторінка «Історія» для звичайного користувача зображена на рисунку 5.10.

№	Автор	Назва	Герой	Архетип	Точність
2	Довженко О.	Україна в огні	Лаврін	герой	92.3%
3	Довженко О.	Україна в огні	Тетяна	наставник	94.1%
4	Довженко О.	Україна в огні	Трохим	герой	86.6%
5	Кобилянська О.	Царівна	Професор	наставник	89.9%
6	Квітка-Оснoв'яненко Г.	Маруся	Маруся	простак	91.2%
7	Квітка-Оснoв'яненко Г.	Маруся	Василь	невинний	93.9%
8	Квітка-Оснoв'яненко Г.	Маруся	Наум	наставник	94.5%

© Copyright 2020, НТУУ "КПІ"

Рисунок 5.10 — Сторінка історії використання системи (користувач)

На сторінці історії класифікацій зображено таблицю, яка містить автора твору, назву, героя, для якого проведена класифікація, визначений архетип, а також точність класифікації моделі.

Для адміністратора сторінка історії має додатковий функціонал, який надає змогу йому редагувати записи у базі даних, такі як назва твору та автор, а також видалення запису. Сторінку «Історія класифікацій» для адміністратора показано на рисунку 5.11.

№	Автор	Назва	Герой	Архетип	Точність
2	Довженко О.	Україна в огні	Лаврін	герой	92.3%
3	Довженко О.	Україна в огні	Олена	простак	94.1%
4	Довженко О.	Україна в огні	Трохим	герой	86.6%
5	Кобиланська О.	Царівна	Професор	наставник	89.9%
7	Квітка-Оснoв'яненко Г.	Маруся	Василь	невинний	93.9%
8	Квітка-Оснoв'яненко Г.	Маруся	Наум	наставник	94.5%

© Copyright 2020, НТУУ "КПІ"

Рисунок 5.11 — Сторінка історії використання системи (адміністратор)

На сторінці історії використання системи адміністратор має дві кнопки навпроти кожного запису, які дозволяють редагувати або видаляти записи. Меню редагування назви твору та автора зображено на рисунку 5.12.



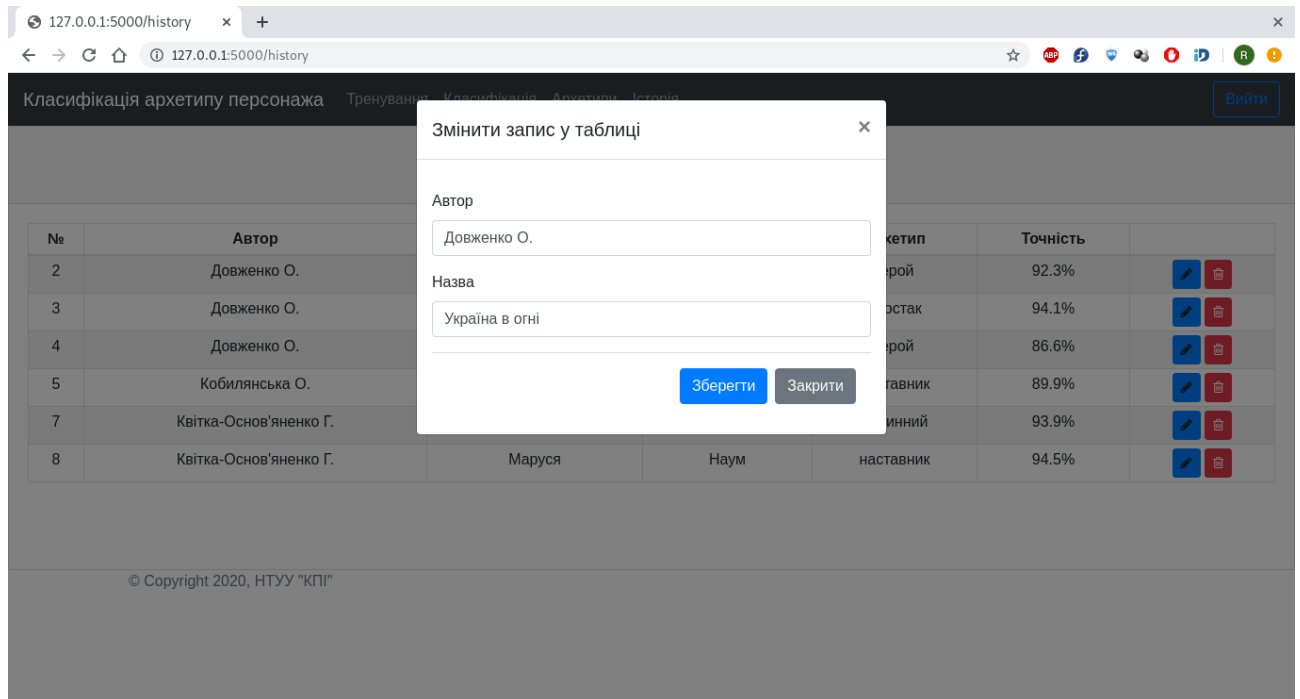


Рисунок 5.12 — Сторінка редагування запису (адміністратор)

## 5.2 Випробування програмного продукту

### 5.2.1 Мета випробувань

Метою випробувань являється перевірка відповідності функцій комплексу задач класифікації архетипу персонажа у літературному творі вимогам технічного завдання.

### 5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

### 5.2.3 Результати випробувань

У даному підрозділі перевірено основний функціонал системи та відповідність роботи системи до поставлених функціональних вимог. Перелік випробувань наведено у таблицях 5.1-5.7.

Таблиця **Error! No text of specified style in document..1** – Тестування роботи тренування

Мета тесту	Перевірка функції «Тренування»
Початковий стан системи	Відкрито меню «Тренування»
Вхідні дані	Датасети у системі
Схема проведення тесту	Вибір датасету з випадуючого меню та натискання на кнопку «Тренування» для проведення тренування моделі
Очікуваний результат	Успішне тренування моделі на основі обраного датасету
Стан системи після проведення випробування	Успішне тренування моделі на основі обраного датасету

Таблиця **Error! No text of specified style in document..2** – Тестування роботи класифікації

Мета тесту	Перевірка роботи меню «Класифікація»
Початковий стан системи	Відкрито меню «Класифікація»

Вхідні дані	Текстовий файл користувача та введене ім'я персонажа у текстове поле
-------------	--

Таблиця **Error! No text of specified style in document..2** (продовження)

Схема проведення тесту	Вибір файлу з дискового носія комп'ютера користувача, введення імені персонажа та натискання кнопки «Визначити»
Очікуваний результат	Успішна класифікація та перехід на сторінку результатів класифікації
Стан системи після проведення випробування	Успішна класифікація та перехід на сторінку результатів класифікації

Таблиця **Error! No text of specified style in document..3** – Тестування виведення результатів класифікації

Мета тесту	Перевірка функції «Результати»
Початковий стан системи	Відкрито меню «Результати»
Вхідні дані	Результати класифікації
Схема проведення тесту	Натискання кнопки «Визначити» у меню класифікації
Очікуваний результат	Виведення назви файлу, імені персонажа, архетипу та точності класифікації моделі
Стан системи після проведення випробування	Виведення назви файлу, імені персонажа, архетипу та точності класифікації моделі

Таблиця **Error! No text of specified style in document..4** – Тестування виведення історії класифікацій

Мета тесту	Перевірка функції «Історія класифікацій»
Початковий стан системи	Відкрито меню «Історія класифікацій»

Таблиця **Error! No text of specified style in document..4** (продовження)

Вхідні дані	Дані з бази про проведену класифікацію: автор та назва твору, ім'я персонажа, архетип, точність класифікації моделі
Схема проведення тесту	Натискання кнопки «Історія» у головному меню
Очікуваний результат	Виведення автора, назви твору, імені персонажа, архетипу та точності класифікації моделі у таблицю
Стан системи після проведення випробування	Виведення автора, назви твору, імені персонажа, архетипу та точності класифікації моделі у таблицю

Таблиця **Error! No text of specified style in document..5** – Тестування авторизації

Мета тесту	Перевірка функції «Авторизація»
Початковий стан системи	Відкрито меню «Авторизація»
Вхідні дані	Логін та пароль, введені в текстове поле користувачем

Мета тесту	Перевірка функції «Авторизація»
Схема проведення тесту	Натискання кнопки «Увійти»
Очікуваний результат	Користувача авторизовано і перенаправлено до сторінки головного меню
Стан системи після проведення випробування	Користувача авторизовано і перенаправлено до сторінки головного меню

Таблиця **Error! No text of specified style in document..6** – Тестування видалення запису адміністратором

Мета тесту	Перевірка функції «Видалити»
Початковий стан системи	Відкрито меню «Історія класифікацій»
Вхідні дані	Користувач авторизований як адміністратор та має доступ до кнопок «Редагувати» та «Видалити»
Схема проведення тесту	Натискання кнопки «Видалити»
Очікуваний результат	Запис видалено з таблиці і бази даних
Стан системи після проведення випробування	Запис видалено з таблиці і бази даних

Таблиця **Error! No text of specified style in document..7** – Тестування редагування запису адміністратором

Мета тесту	Перевірка функції «Редагування»
Початковий стан системи	Відкрито меню «Історія класифікацій»

Мета тесту	Перевірка функції «Редагування»
Вхідні дані	Користувач авторизований як адміністратор та має доступ до кнопок «Редагувати» та «Видалити»
Схема проведення тесту	Натискання кнопки «Редагувати», після появи спливаючого вікна редагування даних та натискання кнопки «Зберегти»
Очікуваний результат	Запис змінюється в таблиці і у базі даних
Стан системи після проведення випробування	Запис змінюється в таблиці і у базі даних

### Висновок до розділу

У даному розділі описано керівництво користувача на прикладі навігації системою та графічного зображення роботи системи. Наведено основну діяльність користувача та результати певних дій. Проведено тестування функціональних вимог методом випробувань, результати описано у відповідних таблицях.

## ЗАГАЛЬНІ ВИСНОВКИ

У дипломному проєкті реалізовано інформаційно-аналітичну систему класифікації архетипу персонажа. Метою системи є спрощення процесу літературного аналізу творів та дослідження залежності архетипів персонажів літературних творів від певних факторів. Задачами розробки є тренування моделі на різних вибірках даних, визначення архетипу введеного персонажа у певному творі, отримання описової характеристики кожного архетипу, отримання історії використання програми. Користувач системи матиме можливість провести класифікацію власного тексту для визначення архетипу обраного персонажа.

У розділі загальних положень описано основні цілі класифікації тексту як напряму у сфері машинного навчання і визначено необхідність розробки поглибленого аналізу літературних творів як для розвитку сфери в цілому, так і для прикладного застосування системи.

У розділі інформаційного забезпечення виконано планування та побудова структури бази даних, описано сутності у вигляді таблиць, наведено графічне зображення структури бази даних.

У розділі математичного забезпечення описано змістовну та математичну постановки, наведено алгоритм розв'язання поставленої задачі.

У розділі програмного та технічного забезпечення описана архітектура системи, вказано засоби розробки, визначено вимоги до технічного забезпечення сервера та клієнта, описано архітектуру системи. Наведено детальний опис класів та функцій програмного продукту, побудовано UML-діаграми.

У технологічному розділі описано керівництво користувача на прикладі навігації системою та графічного зображення роботи системи. Проведено тестування функціональних вимог методом випробувань.

					ДП 6306.00.000 ПЗ	42к.
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ПОСИЛАНЬ

1. Карл Густав Юнг. Архетип і символ. — М., 1991. [Електронний ресурс] — Режим доступу до ресурсу: <https://gtmarket.ru/laboratory/basis/4229/4232>.
2. Carl Gustav Jung, The Archetypes and the Collective Unconscious, 1959. — 320-323 p.
3. GATE [Електронний ресурс] — Режим доступу до ресурсу: <https://gate.ac.uk/>
4. Orange [Електронний ресурс] — Режим доступу до ресурсу: <https://orange.biolab.si/>
5. KNIME [Електронний ресурс] — Режим доступу до ресурсу: <https://www.knime.com/knime-analytics-platform>
6. RapidMiner [Електронний ресурс] — Режим доступу до ресурсу: <https://rapidminer.com/>
7. H. Schuetze, D.A. Hull, J.O. Pedersen, A comparison of classifiers and document representations for the routing problem, in: Proceedings of the Special Interest Group on Information Retrieval, SIGIR Forum; Seattle, WA, The Association for Computing Machinery, New York, 2011 , pp. 229–237.
8. D. Kim, “Prediction performance of support vector machines on input vector normalization methods”, International Journal of Computer Mathematics, 2010. — 459 p.
9. К. В. Воронцов. Лекции по методу опорных векторов — 2007. — 89 с.
10. Beazley D. Python essential reference, 2015. — 734 p.
11. Jack Stuffer, Mastering Flask, 2015. — 288 p.
12. Julian Avila, Scikit-Learn Cookbook — Second Edition, 2017. — pp. 122-130.
13. GitHub [Електронний ресурс] — Режим доступу до ресурсу: <https://www.github.com/>.
14. DigitalOcean [Електронний ресурс] — Режим доступу до ресурсу: <https://www.digitalocean.com/>.



15.nginx [Електронний ресурс] – Режим доступу до ресурсу:  
<https://nginx.org/ru/>.

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК А

**Тексти програмного коду**  
**Інформаційно-аналітична система класифікації архетипу персонажа**

(Найменування програми (документа))

*DVD-R*

(Вид носія даних)

*23 арк, 274 Кб*

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2020 року

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

*app.py*

```
from flask import Flask, Response, render_template, request, redirect, \
    session, url_for, escape, abort
```

```
from sqlalchemy import create_engine
```

```
import sklearn.datasets as skd
```

```
from sqlalchemy.orm import sessionmaker
```

```
from model import Person, Archetype, Book, User
```

```
from flask_login import LoginManager, login_required, \
    login_user, logout_user, current_user
```

```
from sklearn.pipeline import Pipeline
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
from sklearn import svm
```

```
from sklearn.naive_bayes import MultinomialNB
```

```
from security import hash_password, verify_password
```

```
app = Flask(__name__)
```

```
app.secret_key = b"\xec\x18\xd6\x08y\xcb\xa2\r^\xdb\xc4\xf9U\x0fj"
```

```
engine = create_engine('mysql+mysqlconnector://atlas:R3cogn1se!@localhost/archetypes',
    echo=True)
```

```
Session = sessionmaker(bind=engine)
```

```
db_session = Session()
```

```
login_manager = LoginManager()
```

```
login_manager.init_app(app)
```

```
login_manager.login_view = "login"
```

```
categories = [
```

```
    'герой',
```

```
    'наставник',
```

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```
'простак', # everyman
'невинний', # innocent
'злочин',
]
```

```
@app.route('/logout')
@login_required
def logout():
    logout_user()
    return redirect('/')

@login_manager.user_loader
def load_user(user_id):
    return db_session.query(User).filter_by(id=user_id).one()

data_train = skd.load_files('/home/vosiiev/diploma/program/datasets', categories=categories,
encoding='utf-8')
data_test = skd.load_files('/home/vosiiev/diploma/program/test', categories=categories,
encoding='utf-8')
```

```
@app.route('/')
def index():
    return render_template('index.html')

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
```

```

password = request.form['password']
for user in db_session.query(User).all():
    if username == user.username:
        if verify_password(user.password, password):
            login_user(user)
            return redirect(url_for('index'))
        else:
            error = 'Невірно введений пароль'
            return render_template('login.html', error=error)
    else:
        error = 'Користувача з таким іменем не знайдено'
        return render_template('login.html', error=error)
else:
    return render_template('login.html')

```

```
@app.route('/train')
```

```
def train():
```

```
    return render_template('train.html')
```

```
@app.route('/execute', methods=['GET', 'POST'])
```

```
def execute():
```

```
    if request.method == 'POST':
```

```
        text_clf = Pipeline([('vect', TfidfVectorizer()),
                              ('clf', MultinomialNB())
                              ])

```

```
        text_clf.fit(data_train.data, data_train.target)
```

```
        text = []
```

```
        with open('/home/vosiiev/diploma/program/test/наставник/text_3.txt') as f:
```

```
            for line in f:
```

```

        text.append(line)
    predicted = text_clf.predict(text)
    return redirect(url_for('result'))
else:
    return render_template('execute.html')

@app.route('/result', methods=['GET', 'POST'])
def result():
    if request.method == 'POST':
        return redirect(url_for('execute'))
    else:
        return render_template('result.html')

@app.route('/archetypes')
def archetypes():
    return render_template('archetypes.html')

@app.route('/book/edit/<int:id>', methods=['GET', 'POST'])
@login_required
def edit_book(id):
    book = db_session.query(Book).get(id)
    if request.method == 'POST':
        book.author = request.form['author']
        book.name = request.form['bname']

        db_session.commit()

    return redirect(url_for('history'))

```

```
@app.route('/person/delete/<int:id>')
@login_required
def delete_person(id):
    person = db_session.query(Person).get(id)
    db_session.delete(person)
    db_session.commit()
    return redirect(url_for('history'))

@app.route('/history')
def history():
    persons = db_session.query(Person).all()
    return render_template('history.html', persons=persons)

if __name__ == '__main__':
    app.run(debug=True)
```

					ДП 6306.00.000 ПЗ	Док.
Змн.	Арк.	№ докум.	Підпис	Дата		

```
import hashlib, binascii, os
```

```
def hash_password(password):
```

```
    """Hash a password for storing."""
```

```
    salt = hashlib.sha256(os.urandom(60)).hexdigest().encode('ascii')
```

```
    pwhash = hashlib.pbkdf2_hmac('sha512', password.encode('utf-8'),
                                  salt, 100000)
```

```
    pwhash = binascii.hexlify(pwhash)
```

```
    return (salt + pwhash).decode('ascii')
```

```
def verify_password(stored_password, provided_password):
```

```
    """Verify a stored password against one provided by user"""
```

```
    salt = stored_password[:64]
```

```
    stored_password = stored_password[64:]
```

```
    pwhash = hashlib.pbkdf2_hmac('sha512',
                                  provided_password.encode('utf-8'),
                                  salt.encode('ascii'),
                                  100000)
```

```
    pwhash = binascii.hexlify(pwhash).decode('ascii')
```

```
    return pwhash == stored_password
```

*model.py*

```
from sqlalchemy.ext.declarative import declarative_base
```

```
from sqlalchemy import Column, Integer, Float, String, Date, Time, ForeignKey, Boolean, Table,
DateTime
```

```
from sqlalchemy.orm import relationship
```

```
from flask_login import UserMixin
```

					ДП 6306.00.000 ПЗ	Адк.
Змн.	Арк.	№ докум.	Підпис	Дата		



Base = declarative\_base()

class Archetype(Base):

\_\_tablename\_\_ = 'archetype'

id = Column(Integer, primary\_key=True)

name = Column(String(32))

person = relationship('Person', backref='archetype')

class Book(Base):

\_\_tablename\_\_ = 'book'

id = Column(Integer, primary\_key=True)

name = Column(String(32))

author = Column(String(32))

person = relationship('Person', backref='book')

class Person(Base):

\_\_tablename\_\_ = 'person'

id = Column(Integer, primary\_key=True)

name = Column(String(32))

prec = Column(Float)

book\_id = Column(Integer, ForeignKey('book.id'))

archetype\_id = Column(Integer, ForeignKey('archetype.id'))

class User(UserMixin, Base):

\_\_tablename\_\_ = 'user'

id = Column(Integer, primary\_key=True)

username = Column(String(32))

password = Column(String(256))

					ДП 6306.00.000 ПЗ	ЛТ
Змн.	Арк.	№ докум.	Підпис	Дата		

*archetypes.html*

```
{% extends 'base.html' %}
```

```
{% block head %}
```

```
<link rel="canonical" href="https://getbootstrap.com/docs/4.5/examples/carousel/">
```

```
<!-- Bootstrap core CSS -->
```

```
<link href="/docs/4.5/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYYxFfc+NcPb1dKGj7Sk"
crossorigin="anonymous">
```

```
<style>
```

```
.bd-placeholder-img {
  font-size: 1.125rem;
  text-align: middle;
  -webkit-user-select: none;
  -moz-user-select: none;
  -ms-user-select: none;
  user-select: none;
}
```

```
@media (min-width: 768px) {
  .bd-placeholder-img-lg {
    font-size: 3.5rem;
  }
}
```

```
</style>
```

```
<!-- Custom styles for this template -->
```

```
<link href="carousel.css" rel="stylesheet">
```

					ДП 6306.00.000 ПЗ	Док.
Змн.	Арк.	№ докум.	Підпис	Дата		

{ % endblock % }

{ % block body % }

<div class="container marketing">

<br>

<h1><center>Основні архетипи персонажів</center></h1>

<hr class="featurette-divider">

<div class="row featurette">

<div class="col-md-7">

<h2 class="featurette-heading">Герой</h2>

<p class="lead">Герой завжди є основною дійовою особою. Герой переслідує якусь кінцеву мету і повинен зіткнутися і подолати перешкоди на шляху досягнення цієї мети. Він або вона зазвичай морально добрий, хоча це добро, ймовірно, буде оскаржено протягом усієї історії. Здатність героїв залишатися вірними собі, незважаючи на випробування, з якими вони повинні зіткнутися - це те, що робить їх героїчними.</p>

</div>

<div class="col-md-5">



</div>

</div>

<hr class="featurette-divider">

<div class="row featurette">

<div class="col-md-7 order-md-2">

<h2 class="featurette-heading">Наставник</h2>

<p class="lead">Наставник - поширений архетип в літературі. Наставник зазвичай старий, і ця людина часто має якісь магичні здібності або набагато більшу область знань, ніж інші. Наставники допомагають героям під час подорожей, як правило, навчаючи їх, як

					ДП 6306.00.000 ПЗ	Друк.
Змн.	Арк.	№ докум.	Підпис	Дата		

допомагати собі (хоча наставники іноді безпосередньо втручаються в екстремальні ситуації).

Архетип персонажа простака часто виступає в якості "свояка" для аудиторії. Даний архетип персонажа - це просто звичайна людина, але він чомусь повинен зіткнутися з надзвичайними обставинами. Простак може бути головним героєм або опорною фігурою. На відміну від героя, простак не відчуває морального зобов'язання перед своїм завданням; натомість ці персонажі часто опиняються посеред чогось, над чим вони майже не мають контролю. На відміну від героя, даний архетип не намагається робити грандіозних змін або працювати на загальне благо: простаки лише намагаються пройти складну ситуацію у сюжеті.

<div class="col-md-7">

<h2 class="featurette-heading">Невинний</h2>

<p class="lead">Персонажами, що представляють невинний архетип, часто є жінки чи діти. Персонажі даного архетипу всіляко чисті. Незважаючи на здебільшого складні обставини, невинний архетип якось не зазнав корупції та зла інших. Ці архетипи персонажів не дурні: вони просто настільки морально добрі, що злість інших людей не може їх придушити.</p>

</div>

<div class="col-md-5">

<svg class="bd-placeholder-img bd-placeholder-img-lg featurette-image img-fluid mx-auto" width="500" height="500" xmlns="http://www.w3.org/2000/svg" preserveAspectRatio="xMidYMid slice" focusable="false" role="img" aria-label="Placeholder: 500x500"><title>Placeholder</title><rect width="100%" height="100%" fill="#eee"/><text x="50%" y="50%" fill="#aaa" dy=".3em">500x500</text></svg>

</div>

</div>

<hr class="featurette-divider">

<div class="row featurette">

<div class="col-md-7">

<h2 class="featurette-heading">Злодій</h2>

<p class="lead">Злодій хоче зупинити архетип героя від досягнення своєї мети. Лиходій здебільшого є злим, хоча завжди є причина його мотивам. Злодії часто не хочуть нічого іншого, як контролювати і мати владу над усіма і над усім навколо, можливо, тому, що більшість із них таємно сильно мотивовані страхом. Злодії часто є моральною фольгою героя: тобто їх головний порок буде паралельний головній чесноті героя.</p>

</div>

<div class="col-md-5">

<svg class="bd-placeholder-img bd-placeholder-img-lg featurette-image img-fluid mx-auto" width="500" height="500" xmlns="http://www.w3.org/2000/svg" preserveAspectRatio="xMidYMid slice" focusable="false" role="img" aria-label="Placeholder:

					ДП 6306.00.000 ПЗ	Друк.
Змн.	Арк.	№ докум.	Підпис	Дата		

500x500"><title>Placeholder</title><rect width="100%" height="100%" fill="#eee"/><text x="50%" y="50%" fill="#aaa" dy=".3em">500x500</text></svg>

</div>

</div>

<hr class="featurette-divider">

</div>

<!-- /END THE FEATURETTES -->

{ % endblock % }

base.html

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title></title>

<link rel="stylesheet" href="static/css/font-awesome.min.css"/>

<link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"

integrity="sha384-

Vkoo8x4CGsO3+Hh xv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"

crossorigin="anonymous">

<script src="static/js/bootstable.min.js"></script>

<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-

J6qa4849bIE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"

crossorigin="anonymous"></script>

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<script      src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>
<script      src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
integrity="sha384-
wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
crossorigin="anonymous"></script>
<script src="./static/js/editable.js"></script> { % block head % } { % endblock % }
</head>
<body>
  <nav class="navbar navbar-expand-md navbar-dark bg-dark">
    <a class="navbar-brand" href="/">Класифікація архетипу персонажа</a>
    <button    class="navbar-toggler"    type="button"    data-toggle="collapse"    data-
target="#navbarCollapse"    aria-controls="navbarCollapse"    aria-expanded="false"    aria-
label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarCollapse">
      <ul class="navbar-nav mr-auto">
        <li class="nav-item">
          <a class="nav-link" href="/train">Тренування</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/execute">Класифікація</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/archetypes">Архетипи</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/history">Історія</a>
        </li>
      </ul>
    </div>
  </nav>

```

					ДП 6306.00.000 ПЗ	Док.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

</ul>
{ % if not current_user.is_authenticated % }
<a class="btn btn-outline-primary" href="/login">Увійти</a>
{ % else % }
<a class="btn btn-outline-primary" href="/logout">Вийти</a>
{ % endif % }
</div>
</nav>

{ % block body % } { % endblock % }
<footer class="footer">
<div class="container">
<span class="text-muted">&copy; Copyright 2020, НТУУ "КПІ"</span>
</div>
</footer>
</body>
</html>

```

*execute.html*

```

{ % extends 'base.html' % }

{ % block head % }

<script type="text/javascript">
$( '#show_results' ).on( 'shown.bs.modal', function() {
    $( '#myInput' ).focus();
} );
</script>

```

					ДП 6306.00.000 ПЗ	Док.
Змн.	Арк.	№ докум.	Підпис	Дата		



```
{% endblock % }
```

```
{% block body % }
```

```
<br>
```

```
<main role="main" class="container">
```

```
<div class="jumbotron text-center mb-4">
```

```
<h1>Завантажте файл для класифікації</h1>
```

```
<p class="lead">Оберіть *.txt файл з текстом українською мовою та введіть ім'я персонажа</p>
```

```
<form action="/result" method="POST"
```

```
  enctype="multipart/form-data">
```

```
<input type = "file" name = "file" />
```

```
<div class="form-group">
```

```
  <br>
```

```
  <p><label for="exampleInputPassword1">Ім'я</label>
```

```
  <input id="exampleInputPassword1" placeholder=""></p>
```

```
</div>
```

```
<br><br>
```

```
<p><input type="submit" class="btn btn-primary btn-lg" value="Визначити"/></p>
```

```
</form>
```

```
</div>
```

```
</main>
```

```
{% endblock % }
```

*history.html*

```
{% extends 'base.html' % }
```

					ДП 6306.00.000 ПЗ	80к.
Змн.	Арк.	№ докум.	Підпис	Дата		

```
{% block head %}

<link                                rel="stylesheet"                                type="text/css"
href="https://cdn.datatables.net/1.10.20/css/jquery.dataTables.min.css"/>

<script type="text/javascript" src="static/js/jquery.dataTables.min.js"></script>

<script src="https://use.fontawesome.com/3379bb5894.js"></script>
<script type="text/javascript">
    $('#edit_book').on('shown.bs.modal', function() {
        $('#myInput').focus();
    });
</script>

<style media="screen">
    #myInput {
        background-position: 10px 12px; /* Position the search icon */
        background-repeat: no-repeat; /* Do not repeat the icon image */
        width: 100%; /* Full-width */
        font-size: 16px; /* Increase font-size */
        padding: 12px 20px 12px 40px; /* Add some padding */
        border: 1px solid #ddd; /* Add a grey border */
        margin-bottom: 12px; /* Add some space below the input */
    }
</style>

<script type="text/javascript">
    $(document).ready( function () {
        $('#customers').DataTable();
    } );
</script>

{% endblock %}

{% block body %}
```

					ДП 6306.00.000 ПЗ	84к.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<div class="card">
  <h3 class="card-header text-center font-weight-bold text-uppercase py-4">Історія
  класифікацій</h3>
  <div class="card-body">
    <div id="table" class="table-editable">
      <table id="customers" class="table table-bordered table-responsive-md table-striped table-sm
      text-center">
        <caption>
          <div class="p-3">
            </div>
          </caption>
          <thead>
            { % if current_user.is_authenticated % }
            <tr>
              <th onclick="sortTable(0)" class="text-center align-top">№</th>
              <th onclick="sortTable(1)" class="text-center align-top">Автор</th>
              <th onclick="sortTable(2)" class="text-center align-top">Назва</th>
              <th onclick="sortTable(3)" class="text-center align-top">Герой</th>
              <th onclick="sortTable(4)" class="text-center align-top">Архетип</th>
              <th onclick="sortTable(5)" class="text-center align-top">Точність</th>
              <th onclick="sortTable(5)" class="text-center align-top"></th>

            </tr>
          </thead>
          <tbody>
            { % for person in persons % }
            <tr>
              <td class="pt-3-half">{ { person.id } }</td>
              <td class="pt-3-half">{ { person.book.author } }</td>
              <td class="pt-3-half">{ { person.book.name } }</td>
              <td class="pt-3-half">{ { person.name } }</td>
              <td class="pt-3-half">{ { person.archetype.name } }</td>

```

```

<td class="pt-3-half">{{ person.prec }} %</td>
<td class="pt-3-half">
    <a class="btn btn-primary btn-sm" data-toggle="modal" data-
target="#edit_{{ person.book.id }}">
        <i class="fa fa-pencil"></i>
    </a>
    <a class="btn btn-danger btn-sm" href="/person/delete/{{ person.id }}">
        <i class="fa fa-trash-o"></i>
    </a>
</td>
<div id="edit_{{ person.book.id }}" class="modal" tabindex="-1" role="dialog" >
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Змінити запис у таблиці</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                <form method="POST" action="/book/edit/{{ person.book.id }}" novalidate>
                    <p>
                        <div>
                            <label for="content">Автор</label>
                            <input class="form-control" type="text" name="author" id="author"
value="{{ person.book.author }}" required>
                        </div>
                    </p>
                    <p>
                        <div>
                            <label for="content">Назва</label>

```

```

        <input class="form-control" type="text" name="bname" id="bname"
value="{{ person.book.name }}" required>
    </div>
</p>
<div class="modal-footer">
    <input type="submit" class="btn btn-primary" value="Зберегти"></input>
    <button type="button" class="btn btn-secondary" data-
dismiss="modal">Закрити</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</tr>
{% endfor %}
{% else %}
<tr>
    <th onclick="sortTable(0)" class="text-center align-top">№</th>
    <th onclick="sortTable(1)" class="text-center align-top">Автор</th>
    <th onclick="sortTable(2)" class="text-center align-top">Назва</th>
    <th onclick="sortTable(3)" class="text-center align-top">Герой</th>
    <th onclick="sortTable(4)" class="text-center align-top">Архетип</th>
    <th onclick="sortTable(5)" class="text-center align-top">Точність</th>
</tr>
</thead>
<tbody>
    {% for person in persons %}
    <tr>
        <td class="pt-3-half">{{ person.id }}</td>
        <td class="pt-3-half">{{ person.book.author }}</td>
        <td class="pt-3-half">{{ person.book.name }}</td>

```

```
<td class="pt-3-half">{{ person.name }}</td>
<td class="pt-3-half">{{ person.archetype.name }}</td>
<td class="pt-3-half">{{ person.prec }}%</td>
</tr>
```

```
{% endfor %}
```

```
{% endif %}
```

```
</tbody>
```

```
</table>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
{% endblock %}
```

*login.html*

```
{% extends 'base.html' %}
```

```
{% block head %}
```

```
<style>
```

```
.form-signin {
width: 100%;
max-width: 400px;
padding: 15px;
margin: auto;
}
.form-signin .form-control {
```

```
position: relative;
box-sizing: border-box;
height: auto;
padding: 10px;
font-size: 16px;
}
```

```
.form-signin input[type="email"] {
margin-bottom: -1px;
border-bottom-right-radius: 0;
border-bottom-left-radius: 0;
}

.form-signin input[type="password"] {
margin-bottom: 10px;
border-top-left-radius: 0;
border-top-right-radius: 0;
}
```

</style>

{ % endblock % }

{ % block body % }

<div class="text-center mb-4">

<form action="/login" method="POST" class="form-signin">

<br>

<div class="container">

<h1 class="h3 mb-3 font-weight-normal">Авторизація</h1>

<div class="form-label-group">

<p><input type="text" id="username" name="username" class="form-control" placeholder="Лорін"></p>

</div>a

<div class="form-label-group">

					ДП 6306.00.000 ПЗ	Рек.
Змн.	Арк.	№ докум.	Підпис	Дата		

```
<p><input type="password" id="password" name="password" class="form-control"
placeholder="Пароль"></p>
</div>

<button class="btn btn-lg btn-primary btn-block" type="submit">Увійти</button>
</div>
</div>

</form>
{% endblock %}
```

					ДП 6306.00.000 ПЗ	Рек.
Змн.	Арк.	№ докум.	Підпис	Дата		



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”  
Кафедра автоматизованих систем обробки інформації і управління

**УЗГОДЖЕНО**

**Керівник проєкту**

\_\_\_\_\_ Олексій ФІНОГЕНОВ

(підпис)

(вл. ім'я, прізвище)

“13” квітня 2020 р.

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри**

\_\_\_\_\_ Олександр ПАВЛОВ

(підпис)

(вл. ім'я, прізвище)

“14” квітня 2020 р.

ІНФОРМАЦІЙНО-АНАЛІТИЧНА СИСТЕМА КЛАСИФІКАЦІЇ  
АРХИТИПУ ПЕРСОНАЖА

**ТЕХНІЧНЕ ЗАВДАННЯ**

Шифр *ДП 6306.01.000 ТЗ*

на 10 сторінках

Київ – 2020 року

**ЗМІСТ**

<u>1</u>	<u>ЗАГАЛЬНІ ПОЛОЖЕННЯ</u>	4
<u>1.1</u>	<u>ПОВНЕ НАЙМЕНУВАННЯ СИСТЕМИ ТА ЇЇ УМОВНЕ ПОЗНАЧЕННЯ</u>	4
<u>1.2</u>	<u>НАЙМЕНУВАННЯ ОРГАНІЗАЦІЇ-ЗАМОВНИКА ТА ОРГАНІЗАЦІЇ-УЧАСНИКА РОБІТ</u>	4
<u>1.3</u>	<u>ПЕРЕЛІК ДОКУМЕНТІВ, НА ПІДСТАВІ ЯКИХ СТВОРЮЄТЬСЯ СИСТЕМА</u>	4
<u>1.4</u>	<u>ПЛАНОВІ ТЕРМІНИ ПОЧАТКУ І ЗАКІНЧЕННЯ РОБОТИ ЗІ СТВОРЕННЯ СИСТЕМИ</u>	5
<u>2</u>	<u>ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ</u>	6
<u>2.1</u>	<u>ПРИЗНАЧЕННЯ СИСТЕМИ</u>	6
<u>2.2</u>	<u>ЦІЛІ СТВОРЕННЯ СИСТЕМИ</u>	6
<u>3</u>	<u>ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ</u>	7
<u>4</u>	<u>ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</u>	8
<u>4.1</u>	<u>ВИМОГИ ДО ФУНКЦІОНАЛЬНИХ ХАРАКТЕРИСТИК</u>	8
<u>4.2</u>	<u>ВИМОГИ ДО НАДІЙНОСТІ</u>	8
<u>4.3</u>	<u>ВИМОГИ ДО СКЛАДУ І ПАРАМЕТРІВ ТЕХНІЧНИХ ЗАСОБІВ</u>	8
<u>5</u>	<u>СТАДІЇ І ЕТАПИ РОЗРОБКИ</u>	10
<u>6</u>	<u>ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ</u>	11
<u>6.1</u>	<u>ВИДИ ВИПРОБУВАНЬ</u>	11

## 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

### 1.1 Повне найменування системи та її умовне позначення

**Повна назва системи:** Інформаційно-аналітична система класифікації архетипу персонажа.

**Коротке найменування системи:** «Система класифікації архетипу персонажа».

### 1.2 Найменування організації-замовника та організації-учасника робіт

Замовником є кафедра автоматизованих систем обробки інформації і управління Національного технічного університету України "Київський політехнічний інститут ім. Ігоря Сікорського" (далі за текстом — Замовник). Адреса замовника: м. Київ, п. Перемоги 37, 18 корпус ФІОТ.

Розробник сервісу — студент групи ІС-63 кафедри автоматизованих систем обробки інформації та управління Національного технічного університету України "Київський політехнічний інститут ім. Ігоря Сікорського" Восієв Руслан Зійодуллович.

### 1.3 Перелік документів, на підставі яких створюється система

При розробці системи і створення проектно-експлуатаційної документації Виконавець повинен керуватися вимогами наступних нормативних документів:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи. Автоматизовані системи. Стадії створення;
- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем.

					ДП 6306.01.000 ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

#### 1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий термін початку роботи над створенням системи підтримки процесу автомобільних паркувань – 1 грудня 2019 року.

Плановий термін по закінченню роботи над створенням системи класифікації архетипу персонажа – 22 травня 2020 року.

					ДП 6306.01.000 ТЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ

### 2.1 Призначення системи

Даний програмний продукт призначений для вузьких спеціалізацій. Класифікація персонажів за архетипами корисна у сферах, що базуються на літературі, зокрема постановка театральних вистав, створення кіносценаріїв, розробка описових характеристик сюжетів кінокартин, театральних постановок, телепередач, а також психологічних опитувань.

### 2.2 Цілі створення системи

Цілями розробки системи визначення архетипу персонажа літературного твору:

- спрощення процесу літературного аналізу творів;
- візуалізація залежності архетипів персонажів літературних творів від низки факторів (походження твору, автор, жанр).

### 3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Для користування системою користувачеві необхідно перейти на сторінку системи за допомогою веб-браузера.

Перед початком роботи з системою користувач має підготувати файл з текстом твору з розширенням .txt на дисковому носії комп'ютера.

Користувачеві надається можливість скористатись наступними функціями системи класифікації архетипу персонажа:

- авторизація;
- тренування системи;
- додавання власного твору при класифікації;
- класифікація;
- редагування та видалення даних адміністратором.

Для можливості вносити зміни до бази даних користувачеві необхідно авторизуватись як адміністратор.

Об'єктом автоматизації є класифікація архетипу персонажа наданого літературного твору.

					ДП 6306.01.000 ТЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

## 4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вимоги до функціональних характеристик

Актор «Користувач» може обирати тестову вибірку для тренування моделі, проводити класифікацію на основі власного тексту та вибору персонажа, переглядати інформацію про архетипи персонажів, переглядати історію класифікації. Для коректної роботи системи користувач має спершу провести тренування моделі і потім проводити класифікацію.

Актор «Адміністратор» може вносити зміни до бази даних шляхом взаємодії з інтерфейсом.

### 4.2 Вимоги до надійності

Система класифікації архетипу персонажа повинна функціонувати безвідмовно незважаючи на наявність ймовірних дефектів, які можуть проявлятися під час експлуатації. Виправлення таких дефектів повинно виконуватися на етапах розробки, бета-тестування та в процесі введення в дію і в межах промислової експлуатації.

Відмова програмного забезпечення сервісу не повинна призводити до руйнувань даних в інформаційних сховищах.

Будь-які аварійні ситуації повинні бути негайно задокументовані і надіслані розробникам задля усунення причин збою в роботі системи.

### 4.3 Вимоги до складу і параметрів технічних засобів

Для коректної роботи програмного продукту на стороні користувача за наявності наступних умов.

- двохядерний процесор з тактовою частотою не нижче за 1 ГГц;
- 512 МБ оперативної пам'яті;
- 1 ГБ вільного місця на жорсткому диску;
- доступ до мережі інтернет;

- встановлений інтернет браузер (Google Chrome, Firefox, Safari, Opera, IE/Edge.

					ДП 6306.01.000 ТЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		



## 5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

Основні етапи виконання робіт з розробки системи підтримки процесу автомобільних паркувань:

№ з/п	Назва етапу роботи	Термін виконання етапу	Результат виконання
1.	Підготовка технічного завдання на розробку програмного продукту	01.12.2019	
2.	Розробка сценарію роботи	13.12.2019	
3.	Технічне проектування – функціональність, модулі, задачі, цілі тощо	24.12.2019	
4.	Узгодження з керівником інтерфейсу користувача	10.01.2020	
5.	Розробка інформаційного забезпечення	20.01.2020	
6.	Розробка програмного забезпечення	07.02.2020	
7.	Налагодження програми	07.04.2020	
8.	Тестування програми	17.04.2020	
9.	Здача готового програмного продукту замовнику	22.05.2020	

## 6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ

### 6.1 Види випробувань

Задля перевірки правильності роботи програмного продукту будуть проведені випробування на основі таких документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

У ході випробування буде використано такі методи перевірки:

- функціональне тестування;
- тестування роботи бази даних.

У якості функціонального тестування системи класифікації архетипу персонажа буде проведено перевірку відповідності програмного продукту до визначених функціональних вимог. Необхідна перевірка наступних функцій:

- авторизація;
- тренування системи;
- додавання власного твору при класифікації;
- класифікація;
- редагування та видалення даних адміністратором.

Також необхідним є тестування системи на валідацію заповнених полів на випадок введення некоректних даних користувачем.

У якості тестування роботи бази даних буде проведено тестування збереження даних про виконані класифікації на сервері.



Власник документу:  
Попенко Володимир Дмитрович

Дата перевірки:  
08.06.2020 18:21:53 EEST

Дата звіту:  
09.06.2020 16:34:45 EEST

ID перевірки:  
1003881985

Тип перевірки:  
Doc vs Internet + Library

ID користувача:  
77149

Назва документу: Vosiev\_bachelor\_is63

ID файлу: 1003896784 Кількість сторінок: 48

Кількість слів: 7077 Кількість символів: 53365 Розмір файлу: 95.43 KB

## 6.75% Схожість

Найбільша схожість: 3.26% з джерело бібліотеки. ID файлу: 5839560

2.67% Схожість з Інтернет джерелами

1

Page 50

6.75% Текстові збіги по Бібліотеці акаунту

16

Page 50

## 0.78% Цитат

Цитати

2

Page 51

Вилучення переліку посилань вимкнено

## 0% Вилучень

Вилучений текст відсутній

## Підміна символів

Заміна символів

39



ДП 6306.00.000 ПЗ

## ВСТУП

Комп'ютерна лінгвістика як науковий напрям існує уже майже 70 років. Поява перших мов програмування, а також поступовий розвиток інформаційних технологій дав поштовх для розвитку машинного перекладу текстів, особливо радянських наукових журналів. Хоча і такий переклад значно поступався і поступається у якості «живому», розвиток напрямку є значним і заслуговує на окрему увагу.

У сучасному світі для повсякденного користувача інтернету не є новиною можливість перекладати текст онлайн, виконувати миттєвий пошук слів, або визначати автора на назву тексту за уривком. Більш того, за останні роки використання систем штучного інтелекту стало нормою для виконання прикладних лінгвістичних завдань. Одним з таких завдань є класифікація тексту.

Існує велика кількість параметрів класифікації тексту, як і методів автоматизації даної задачі шляхом програмування. Проте, якщо для глобальних завдань знайдено велику кількість можливих рішень, для більш детального аналізу тексту їх значно менше.

Класифікація архетипу персонажа є новою задачею комп'ютерної лінгвістики, адже досі не було створено відповідних методів її вирішення. Персонажі літературних творів здебільшого виступають у ролі шаблонних фігур з певними індивідуальними рисами і кожен з таких шаблонів має певні особливості, які відрізняють їх один від одного. Автоматизація визначення характерних ознак героїв шляхом програмування є ключовим завданням.

Практичне значення одержаних результатів. Розроблено алгоритм класифікації архетипу персонажа літературного твору.

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

ДП 6306.00.000 ПЗ

## 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

## 1.1 Опис предметного середовища

Предметним середовищем є процес класифікації текстів на основі інформації про окремо взятих персонажів з попереднім тренуванням моделі на певній вибірці даних.

Автором класифікації людей за архетипами є Карл Густав Юнг [1]. Він був психологом і його ідея полягала у тому, що кожна людина виступає в якості носія колективного несвідомого — того, що представляє собою спільні особливості для різних осіб.

На основі вчень Юнга створено велику кількість архетипів, проте наведена класифікація є основною — 5 типів, символізуючих людську мотивацію.

Список основних архетипів наведено нижче [2]:

- невинний — розвиток віри, впевненості й оптимізму;
- простак — усвідомлення існування поганих речей, розвиток реалізму;
- герой — навчання змагання, досягнення цілей і захист себе;
- наставник — демонстрація турботи і співчуття до інших, допомога;
- злодій — мета завадити героєві досягти своєї мети.

Даний розподіл використовується і для героїв літературних творів, адже вони здебільшого є прототипами реальних людей.

## 1.1.1 Опис процесу діяльності

Для проведення літературного аналізу певного твору науковцеві-літературознавцю необхідно витратити велику кількість часу. Прочитання більшості літературних творів у звичайної людини займає десятки годин, якщо це роман чи повість. Тим більше, одного читання не достатньо для

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ДП 6306.00.000 ПЗ

проведення класифікації тексту та достовірного аналізу його змісту. Враховуючи тематику даного проекту, мова йде про аналіз придатності літературного персонажа до певного архетипу.

Автоматизація роботи полягає перш за все в економії часу на аналіз. По-друге, сучасні технології надають можливість виконувати завдання класифікації з точністю близько до 1, в залежності від розміру тренувальної вибірки.

Сам аналіз полягає у неодноразовому прочитанні твору літературознавцем, визначенні абзаців, що містять опис персонажа, його взаємодію з іншими дійовими особами твору, розвиток від початку розповіді до її розв’язки.

1.1.2 Опис функціональної моделі

Даний проект розроблюється для дослідницьких цілей, а саме аналізу роботи та використання алгоритмів класифікації для більш вузької спеціалізації, і передбачає взаємодію з одним користувачем, тому актор один.

						ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			8



ДП 6306.00.000 ПЗ

Таблиця 1.1 — Функціональні вимоги

Актор	Варіант	Функціональна вимога	Пріоритет	використання
Користувач	Головне меню	<div>— Система надає високий короткий опис роботи програми.</div> <div>— Система надає можливість скористатись кнопкою «Почати роботу» для переходу на сторінку тренування моделі.</div> <div>— Система надає можливість користувачеві скористатись меню вгорі для швидкого переходу між сторінками у системі.</div>		
Адміністратор	Головне меню	<div>— Система надає можливість авторизуватись адміністраторові за допомогою кнопки «Увійти».</div> <div>— Система надає можливість авторизованому користувачеві вийти з облікового запису за допомогою кнопки «Вийти».</div>		

Таблиця 1.1 (продовження)

Змн.

Арк.

№ докум.

Підпис

Дата

ДП 6306.00.000 ПЗ

Арк.

9

ДП 6306.00.000 ПЗ				
Користувач	Додавання датасету	— Система надає можливість Високий користувачеві обрати датасет для тренування моделі. — Система надає можливість користувачеві обирати датасети з різних жанрів літератури за допомогою випадаючого меню.		
Користувач	Тренування моделі	— Система надає можливість Високий користувачеві за натисканням кнопки «Тренування» виконати тренування моделі на обраному датасеті.		
Користувач	Введення даних для класифікації	— Система надає можливість Високий користувачеві обрати файл з дискового носія на комп'ютері. — Система надає можливість користувачеві ввести у текстове поле ім'я персонажа.		
Користувач	Визначення архетипу персонажа	— Система надає можливість Високий користувачеві натиснути кнопку «Визначити» та.  — Система має зберігати виведений результат у базу даних.		
Таблиця 1.1 (закінчення)				
ДП 6306.00.000 ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата

ДП 6306.00.000 ПЗ				
Користувач	Перегляд результатів класифікації виведення даних про результати	— Система надає можливість користувачеві отримати класифікації на екран.	Високий	
		— Система надає можливість натиснути кнопку «Повернутись» для повернення до меню класифікації.		
Користувач	Перегляд інформації про архетипи	— Система надає можливість користувачеві натиснути елемент меню «Архетипи» для перегляду інформації про основні архетипи в літературі.	Низький	
Користувач	Перегляд історії запусків класифікації інформацію з попередніх запусків системи.	— Система надає можливість користувачеві переглянути	Низький	
Адміністратор	Перегляд та редагування адміністраторові історії запусків класифікації	— Система надає можливість змінювати записи у базі даних за допомогою таблиці на сторінці.	Високий	
<div>1.2 Огляд наявних аналогів</div> <p>На сьогодні існує велика кількість різноманітних сервісів, що надають змогу користувачеві виконувати інтелектуальний аналіз тексту — процес автоматичного аналізу звичайних неструктурованих текстових документів комп'ютером з метою витягнення високоякісної структурованої інформації.</p>				
Змн.	Арк.	№ докум.	Підпис	Дата
ДП 6306.00.000 ПЗ				Арк. 11

ДП 6306.00.000 ПЗ

Більшість з них являє собою десктопні застосунки, або бібліотеки певних мов програмування. Розглянемо деякі з них.

GATE [3] (General Architecture for Text Engineering — основна архітектура для обробки тексту) — масштабний програмний продукт з відкритим кодом, який включає в себе інструменти для підтримки життєвого циклу ПЗ — від проектування і розробки до спільного використання великою кількістю серверів у цілях анотації документів. Також даний інструмент надає інтерфейс для застосунків всередині організації за допомогою бібліотеки об'єктів Java.

Серед переваг GATE не тільки розвинена екосистема, а також двадцятирічний досвід існування на ринку та переклад на десятки мов. З практичних переваг — обробка документів не перевантажує пам'ять, оскільки виконується послідовно, проте швидкість роботи системи при цьому значно падає.

Orange Інструмент для інтелектуального аналізу Orange [4] включає в себе розширення для роботи з неструктурованими масивами даних — в тому числі, з текстами. При цьому широкі можливості візуалізації Orange використовуються в цілях text mining. Подібні розширення зручні тим, що можна не відмовлятися від звичного ПЗ, коли з'являються нові задачі. Зручний графічний користувацький інтерфейс та інструменти візуального програмування роблять Orange привабливим для користувачів, проте його «зав'язка» на Python може сподобатись не кожному.

Система інтелектуального аналізу даних KNIME [5] містить плагін для обробки тексту з назвою KNIME Text Processing. Плагін, як і уся система, має відкритий код і підтримує шестиступеневий процес обробки тексту — від читання до синтаксичного аналізу через розпізнавання сутностей, фільтрації та маніпуляції до підрахунку кількості слів, виділення ключових понять та візуалізації. Все це дає користувачеві широкі можливості роботи з текстом,

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

ДП 6306.00.000 ПЗ

проте він може знайти у KNIME пару вагомих недоліків. Наприклад, система не читає дані з MS Excel і не працює з технологією OLAP.

Розширення для роботи з текстами відомого інструменту RapidMiner [6] сподобається консерваторам. Воно відрізняється широкими можливостями у вирішенні завдань з text mining. Наголос робиться на статистичному аналізі, а дані вивантажуються з безлічі популярних форматів — в тому числі, PDF.

Безсумнівний плюс — графічний інтерфейс, що дозволяє управляти потоками даних, буквально просто «перетягуючи» їх з місця на місце. Правда, разом з перевагами, аддон успадкував і недоліки «батьківської системи» — обмежений обсяг даних, які можна обробити за допомогою безкоштовної версії ПЗ. Правда, для тексту один гігабайт — цифра чимала (згадати хоча б, що «Війна і мир» у форматі txt займає всього 300 КБ пам'яті), тож як мінімум в академічних колах цього недоліку можуть взагалі не помітити.

### 1.3 Постановка задачі

#### 1.3.1 Призначення розробки

Призначення розробки — визначення архетипу персонажа літературного твору.

#### 1.3.2 Цілі та задачі розробки

Цілями розробки системи визначення архетипу персонажа літературного твору:

- спрощення процесу літературного аналізу творів.

Задачі розробки:

- тренування моделі на різних вибірках даних;
- визначення архетипу введеного персонажа у певному творі;

						ДП 6306.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата			13



ДП 6306.00.000 ПЗ				
<ul style="list-style-type: none"><li>— визначення метрик результатів виконання машинного навчання на тестовій вибірці;</li><li>— отримання описової характеристики кожного архетипу;</li><li>— отримання історії використання програми;</li></ul> <p>Висновок до розділу</p> <p>На основі проведеного порівняння існуючих систем інтелектуального аналізу даних знайдено проблему відсутності аналогів розроблюваної системи. Було вивчено основні цілі класифікації тексту як напряму у сфері машинного навчання і визначено необхідність розробки поглибленого аналізу літературних творів як для розвитку сфери в цілому, так і для прикладного застосування системи.</p>				

Джерел на сторінці: 1-21, 39

Сторінка 11 з 53

ДП 6306.00.000 ПЗ					
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ					
2.1 Вхідні дані					
Вхідними даними даної системи є:					
— дані, що надходять від користувача;					
— дані, що містяться у базі, необхідні для тренування моделі.					
Нижче наведено детальний розгляд даних.					
Дані від користувача:					
— обраний з системи користувача файл з розширенням *.txt, який містить літературний твір;					
— введене користувачем в текстове поле ім'я персонажа твору.					
Дані у базі, необхідні для тренування моделі:					
— тексти літературних творів;					
— імена та архетипи літературних героїв.					
2.2 Вихідні дані					
Вихідними даними є:					
— архетип персонажа у літературному творі;					
— описова характеристика наведеного архетипу.					
2.3 Опис структури бази даних					
На стадії проектування бази даних було виділено сутності «Архетип», «Текст», «Персонаж» та «Користувач». База даних складається з 4-х таблиць. Графічне зображення схеми бази даних знаходиться у графічних матеріалах.					
Сутності схеми описані у таблицях 2.1-2.3					
Сутність «Текст» заходиться у таблиці 2.1. Таблиця містить поля id, title, author, text — дані про текст для тренування моделі.					
					Арк.
ДП 6306.00.000 ПЗ					15
Змн.	Арк.	№ докум.	Підпис	Дата	

Джерел на сторінці: 3-7, 14, 19, 24, 34

Сторінка 12 з 53



ДП 6306.00.000 ПЗ					
Таблиця 2.1 — Опис таблиці «Text»					
Код	Тип	Обов'язкове	Унікальне	Первинний ключ	Зовнішній ключ
	Опис даних				
	ідентифікатор тексту	integer	X	X	X
	Назва тексту	nvarchar(32)	X		
	Автор тексту	nvarchar(32)	X		
	Текст	nvarchar(MAX)	X		
Сутність «Архетип» заходиться у таблиці 2.2. Таблиця містить поля id, name, descr — дані про архетипи літературних персонажів для тренування та практичного використання моделі.					
Таблиця 2.2 — Опис таблиці «Archetype»					
Код	Тип	Обов'язкове	Унікальне	Первинний ключ	Зовнішній ключ
	Опис даних				
ідентифікатор архетипу	integer	X	X	X	
ДП 6306.00.000 ПЗ					Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	16



ДП 6306.00.000 ПЗ

Таблиця 2.2 (продовження)

n ame	Назва архетипу	nvarchar( 32)	X			
d escr	Опис архетипу	nvarchar( MAX)				

Сутність «Персонаж» заходиться у таблиці 2.3. Таблиця містить поля id, name, text\_id, arch\_id — дані про літературних персонажів для тренування та практичного використання моделі, а також зовнішні ключі, які вказують на архетип персонажа та текст, з якого персонаж походить.

Таблиця 2.3 — Опис таблиці «Character»

К од	Опис	Тип даних		Обов'язкове	Унікальне	Первинний ключ	Зовнішній ключ
i d	Ідентифікатор персонажа	integer	X	X	X		
n ame	Ім'я персонажа	nvarchar( 32)	X				
te xt_id	Зовнішній ключ тексту	integer	X				X
a rch_id	Зовнішній ключ архетипу	integer					X


Арх



ДП 6306.00.000 ПЗ

Сутність «Користувач» заходиться у таблиці 2.4. Таблиця містить поля id, username, password — дані користувача, необхідні для авторизації у системі.

Таблиця 2.4 — Опис таблиці «User»

Код	Опис	Тип даних	Обов'язковий	Унікальний	Первинний ключ	Зовнішній ключ
id	Ідентифікатор персонажа	integer	X	X	X	
username	Логін користувача	nvarchar(32)	X			
password	Пароль користувача	nvarchar(256)	X			

## Висновок до розділу

У даному розділі виконано планування та побудова структури бази даних, описано сутності у вигляді таблиць, наведено графічне зображення структури бази даних.

~~ДП 6306.00.000 ПЗ~~

Арк.

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

18

ДП 6306.00.000 ПЗ

### 3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

#### 3.1 Змістовна постановка задачі

Класифікація даних [7] – загальна задача машинного навчання. Нехай деякі задані спостереження (об'єкти) належать до одного з двох класів. Завдання полягає в тому, щоб визначити, до якого класу будуть належати нові спостереження.

Аналогічно, класифікація тексту — це одна із задач інформаційного пошуку, яка полягає у відношенні документу до одної з декількох категорій на основі змісту документа.

Найчастіше класифікація текстів використовується для таких задач:

- перевірка на плагіат;
- вилучення спаму;
- визначення мови тексту;
- таргетування реклами;
- розподіл каталогів за жанрами.

На початку 2000-х років класифікація текстів була однією з часто обговорюваних тем, проте після винаходу підходу TF-IDF, який показує точність 95%-99.9%, ажіотаж стих. За такої точності слід більше звернути увагу саме на дані, які оброблюються, а не алгоритм.

У межах даного проекту розглядається особливий варіант класифікації за архетипами персонажів.

Користь даного виду класифікації не є такою ж очевидною, як, наприклад, класифікація електронних листів, каталогів, судових рішень тощо. Проте, класифікація літератури за персонажами є доволі розповсюдженою у вузьких спеціалізаціях. Одними з таких є:

- постановка театральних вистав;
- розробка кіносценаріїв;

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

ДП 6306.00.000 ПЗ

- розробка описових характеристик сюжетів кінокартин, театральних постановок, телепередач;
- психологічні опитування.

При постановці кіно- та театральних сюжетів велику роль відіграють персонажі, тому для поглибленого розуміння мотивів та характерів персонажів буде актуальною розгляд їх психологічних архетипів.

Оскільки виділені архетипи засновані психологом і були створені саме для класифікації людських мотивів, тому і їх використання у психології є очевидним. На основі даних архетипів можна створювати психологічні тести, за результатами яких людина може сформулювати певне уявлення про особливості власної психології, які раніше не були очевидними.

У даному проекті на основі алгоритмів машинного навчання та класифікації текстів необхідно розробити модель, яка обробляє вхідні текстові дані і виконує безпосередньо класифікацію літературного твору з метою знаходження архетипу введеного користувачем персонажа.

### 3.2 Математична постановка задачі

Нехай  $X$  — простір об'єктів (наприклад,  $\mathbb{R}^n$ ),  $Y$  — кінцева множина імен класів (у даному випадку назви архетипів). Існує невідома цільова залежність — відображення  $y^*: X \rightarrow Y$ , значення якої відомі лише на об'єктах кінцевої тренувальної вибірки  $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$ . Необхідно побудувати алгоритм:  $a: X \rightarrow Y$ , здатний класифікувати довільний об'єкт  $x \in X$ .

### 3.3 Обґрунтування методу розв'язання

Метод опорних векторів (англ. Support Vector Machine, SVM) [8] - один з найбільш популярних методів навчання, який застосовується для розв'язання задач класифікації і регресії. Основна ідея методу полягає в знаходженні гіперплощини в  $N$ -мірному просторі ( $N$  — кількість ознак), що

						ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			20

ДП 6306.00.000 ПЗ

розділяє об'єкти вибірки оптимальним способом. Алгоритм працює в припущенні, що чим більше відстань між розділяючою гіперплощиною і об'єктами поділюваних класів, тим меншою буде середня похибка класифікатора.

Гіперплощини – це такі площини, що допомагають класифікувати точки даних. Точки даних, що падають з будь-якої сторони гіперплощини, можна віднести до різних класів. Також розмірність гіперплощини залежить від кількості ознак. Якщо кількість вхідних функцій дорівнює 2, то гіперплощина — це всього лінія. Якщо кількість вхідних ознак дорівнює 3, то гіперплощина стає двовимірною площиною. Візуалізація гіперплощин розмірності 3 і більше є набагато складнішою. Графічне зображення гіперплощин розмірності 1 та 2 зображено на рисунку 3.1.

Рисунок 3.1 — Гіперплощини з розмірністю 1 та 2 відповідно

Опорні вектори — це точки даних, які знаходяться ближче до гіперплощини і впливають на її положення та орієнтацію [9].. Використовуючи ці вектори, максимально збільшується запас класифікатора. Видалення опорних векторів змінить положення гіперплощини. Поняття опорних векторів та гіперплощин є основними при розробці алгоритму методом опорних векторів.

Одна з переваг SVM-методу в тому, що так як він намагається визначити оптимальний напрямок поділу простору ознак, розглядаючи комбінації ознак, він досить стійкий до великих розмірностей. Текстові дані ідеально підходять для класифікатора SVM через розрідженість даних великої розмірності, що є відмінною рисою текстів.

Недоліки методу: складна інтерпретованість параметрів алгоритму і нестійкість по відношенню до «викидів» у вихідних даних.

Арк.

ДП 6306.00.000 ПЗ

Змн. Арк. № докум. Підпис Дата

21





ДП 6306.00.000 ПЗ

3.4 Опис методів розв’язання

Оберемо пряму, максимально далеко проходить від точок. Відстань від неї до найближчої точки з кожного боку максимальна.

Якщо така пряма існує, то її називають гіперплощиною максимальної різниці, і лінійний класифікатор, що її визначає, відповідно класифікатор максимальної різниці або перцептрон оптимальної стабільності (стійкості).

Опорні вектори — це точки, для яких відстань до гіперплощини

$\frac{1}{2}|w|V$

1. Вони є ефективними елементами на навчальній вибірці.

Графічне зображення опорних векторів представлено на рисунку 3.2.

Рисунок 3.2 — Візуалізація опорних векторів

Важливо зазначити, що якщо всі точки, крім опорних векторів, видалити, алгоритм методу опорних векторів залишиться колишнім. Ця властивість робить SVM унікальним, несхожим на всі інші методи, такі як kNN, LLSF, NNet і NB, де всі точки тренованої вибірки використовуються для оптимізації функції. Теоретична відмінність призводить до значної розбіжності між SVM та іншими методами на практиці.

Цей підхід узагальнюється на багатовимірний випадок.

У даному проекті використовується лінійний метод опорних векторів для виконання класифікації, проте слід описати і нелінійний варіант для розуміння різниці між ними у загальному вигляді.

Лінійний метод опорних векторів

Дано спостереження для навчання D, набір складається з n об’єктів:

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

ДП 6306.00.000 ПЗ

$$D = \{(x_i, y_i) \mid \forall x_i \in R^p, y_i \in \{-1, 1\}\}_{i=1}^n \quad (3.1)$$

де  $y$  приймає значення  $-1$  або  $1$ , визначаючи, до якого класу належить кожна точка  $x_i$ . Кожна точка  $x_i$  — вектор розмірності  $p$ .

Необхідно знайти гіперплощину максимальної різниці, яка розділяє спостереження, що мають  $y_i=1$  від об'єктів  $y_i=-1$ .

Будь-яку гіперплощину можна записати як безліч точок  $x$ , що задовольняють:

$$w * x - b = 0 \quad (3.2)$$

де  $*$  - скалярний добуток нормалі до гіперплощини на вектор  $x$ .

Параметр  $\frac{|w|}{b}$  визначає зміщення гіперплощини від початку координат вздовж нормалі  $w$ .

Якщо навчальні дані є лінійно нероздільні, можна вибрати дві паралельні гіперплощини таким чином, що вони розділять безліч точок на 2 класи, і точок між ними не буде.

Потім намагаються максимізувати дистанцію між ними, одночасно роблячи поворот і паралельно зрушення паралельних прямих.

Область, обмежена 2 гіперплощинами, називається різницею (маржею).

Ці гіперплощини можуть бути описані рівняннями:

$$w * x - b = 1 \quad (3.3)$$

$$w * x - b = -1 \quad (3.4)$$

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

ДП 6306.00.000 ПЗ				
<p>Знаходимо відстань між гіперплощинами: <math>\frac{ w V}{\epsilon}</math>.</p> <p>Для того, щоб дистанція була максимальною, мінімізуємо <math> w V</math>.</p> <p>Щоб виключити всі точки зі смуги, ми повинні переконатися, що для всіх спостережень справедливим є:</p> $w \cdot x_i - b \geq 1, \text{ для } x_i \text{ з першого класу} \tag{3.5}$ $w \cdot x_i - b \leq -1, \text{ для } x_i \text{ з другого класу} \tag{3.6}$ <p>Еквівалентно для <math>0 \leq i \leq n</math> :</p> $y_i (w \cdot x_i - b) \geq 1 \tag{3.7}$ <p>Далі вирішується задача оптимізації:</p> $ w V \rightarrow \min \tag{3.8}$ $y_i (w \cdot x_i - b) \geq 1, \text{ для } 0 \leq i \leq n \tag{3.9}$ <p>Нелінійний метод</p> <p>Для створення нелінійного класифікатора використовується довільна функція ядра. Кожний скалярний добуток замінюється на нелінійну функцію ядра. Це дозволяє знаходити гіперплощину максимальної різниці в трансформованому просторі функцій.</p> <p>Зміна може бути нелінійною і трансформуватися в простір з більш високою розмірністю.</p>				
				ДП 6306.00.000 ПЗ
Змн.	Арк.	№ докум.	Підпис	Дата
				24

Джерел на сторінці: 1-19, 30, 55, 57

Сторінка 21 з 53

ДП 6306.00.000 ПЗ

Незважаючи на те, що класифікатор є гіперплощиною в багатовимірному просторі функцій, він може бути нелінійним в вихідному просторі навчальної вибірки.

Багатокласовий метод опорних векторів

Найбільш популярний підхід — перехід від завдання класифікації на множину класів до множинної задачі розбиття на два класи. Даний метод використовує 2 стратегії:

- один проти всіх;
- один проти одного.

У першому випадку навчається  $N$  класифікаторів, де  $N$  — кількість класів. Класифікатор з найвищим значення функції виходу привласнює новий об'єкт до певного класу.

У випадку «один проти одного» також навчається  $N$  класифікаторів, тільки в цьому разі об'єкт присвоюється до того класу, до якого його віднесло більшість класифікаторів.

Висновок до розділу

У даному розділі було описано змістовну та математичну постановку, наведено алгоритм розв'язання поставленої задачі.

						ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			25

ДП 6306.00.000 ПЗ

## 4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

## 4.1 Засоби розробки

Розробка системи проведена на базі технології Python [10]. Дана мова програмування є високорівневою і добре підходить для виконання задач, пов'язаних безпосередньо з веб-застосуваннями та системами машинного навчання й штучного інтелекту. Технологія не дає такого ж виграшу у швидкості роботи, як такі низькорівневі мови, як С або С++. Проте розробникові надається змога зосередитись на алгоритмах програми, не витрачаючи велику кількість часу на написання коду.

Найбільша перевага технології Python — це її зовнішні бібліотеки. Вони значно розширюють можливості мови програмування і є невід'ємною частиною Python. Одними з таких є Flask [11] та scikit-learn [12].

Flask — веб-фреймворк на базі Python. Основною його перевагою є його простота розробки застосувань та взаємодія з клієнтською частиною. Інтерфейс даного проекту повністю розроблено за допомогою вищевказаного фреймворка.

Scikit-learn — бібліотека, що надає корисні інструменти для розробки моделей машинного навчання. У даній системі необхідно якісно оброблювати текст для правильної роботи моделі та отримання достовірних результатів класифікації, тому sklearn є одним із допоміжних методів, використаних при розробці системи.

Локальний сервер, використаний для розробки, є на базі операційної системи Fedora 31, що є однією з розширень Unix-систем. Операційні системи на базі Linux широко використовуються розробниками у світі через свою низькорівневність. Контроль над усією системою може відбуватись за допомогою терміналу, тобто такі операції як запуск сервера, робота з

						ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			26

ДП 6306.00.000 ПЗ

системою управління базою даних та системою контролю версій здійснюються в одному застосуванні.

Веб-сервер розгорнуто на базі операційної системи Ubuntu 16.04 за допомогою онлайн-сервісу DigitalOcean. Недоліком даного сервісу є те, що він не безкоштовний, на відміну від таких аналогів як Heroku.

Створення діаграм виконано за допомогою онлайн-сервісу draw.io. Дана платформа не вимагає скачування десктоп-застосування, має зручний та зрозумілий інтерфейс, а також паттерни більшості UML-діаграм, необхідних для розробки програмного продукту.

У якості системи контролю версій обрано Git, платформа — GitHub. Використання системи контролю версій є загально прийнятим підходом до розробки програмного забезпечення через можливість зберігати прогрес виконання роботи, незалежно від стану машини, на якій проводилась розробка. Також розгортання веб-застосування вимагає використання Git для зручного оновлення функціоналу системи.

## 4.2 Вимоги до технічного забезпечення

### 4.2.1 Загальні вимоги

Для правильної роботи системи класифікації та тренування моделі необхідний сервер з наступними мінімальними параметрами:

- процесор з частотою 1,6 ГГц або вище;
- об'єм оперативної пам'яті 2 ГБ або більше;
- жорсткий диск об'ємом не менше 20 ГБ;
- операційна система Linux (Fedora 31, Ubuntu 16.04 або 18.04).

Для користування системою на стороні клієнта мають бути виконані наступні вимоги:

- наявність веб-браузера Google Chrome, Firefox, Microsoft Edge, Opera.

						ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			27





ДП 6306.00.000 ПЗ

4.3 Архітектура програмного забезпечення

Система класифікації архетипу персонажа літературного твору є веб-застосуванням з використанням алгоритмів машинного навчання, тому для її розробки використано мову програмування Python.

Класифікація архетипу персонажа відбувається у декілька етапів. Перший етап — збір тренувальних датасетів для моделі. Оскільки для тренування системи розпізнавати архетип персонажа необхідно спочатку вказати в тренувальних даних, які бувають архетипи та навести приклади, усі датасети зібрано «вручну» шляхом аналізу літературних творів та вказуванням архетипів основних дійових осіб. Кожен датасет містить список

з імен героїв та їх архетипів на початку файла. Вигляд датасетів у системі зображено на рисунку 4.1.

Рисунок 4.1 — Датасет української прози

Приклад одного файлу із датасету зображено на рисунку 4.2.

Рисунок 4.2 — Пантелеймон Куліш — Чорна Рада

Для тренування моделей було обрано датасети з української та зарубіжної літератури. Як для української, так і для зарубіжної було обрано більше 50 творів по 2-5 персонажів у кожному. Використано роботи 22-х авторів української літератури та 12-х зарубіжної.

Після підготування датасету програмі необхідно його обробити для виконання класифікації. Спочатку виконується обробка тексту на знаходження абзаців з участю вказаних персонажів, потім формуються нові

						ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			28

ДП 6306.00.000 ПЗ

тексти окремо для кожного персонажа і саме вони беруть участь у тренуванні моделі на встановлення архетипів.

Відфільтровані тексти зчитуються програмою і відбувається перетворення тексту — векторизація. Вона складається з п'яти етапів:

- токенізація — розбиття довгих ділянок тексту на більш дрібні (абзаци, пропозиції, слова);
- нормалізація — приведення тексту до «очищеного» вигляду (єдиний реєстр слів, відсутність знаків пунктуації, розшифровані скорочення, словесне написання чисел і т.д.). Це необхідно для застосування уніфікованих методів обробки тексту;
- стеммізація — приведення слова до його кореня шляхом усунення суфіксів, приставок, закінчень тощо;
- лематизація — приведення слова до смислової канонічної форми слова (інфінітив для дієслова, називний відмінок однини - для іменників і прикметників);
  - чистка — видалення стоп-слів, які не несуть смислового навантаження (артиклі, займенники, сполучники, прийменники і т.д.).

Хоча мова програмування Python надає готові рішення для даної задачі, скористатись ними неможливо через використання української мови у датасетах та розробці. Готові алгоритми не розпізнають кодування і методи не виконуються як слід. Тому алгоритми для виконання вищевказаних дій розроблено власноруч саме для української мови.

Наступним кроком є застосування алгоритму машинного навчання, опис якого наведено у розділі 3.

Для взаємодії з користувачем обрано архітектуру системи на основі веб-застосунку.

Веб-застосунок — це клієнт-серверне рішення, у якому клієнт виконує взаємодію з сервером за допомогою веб-браузера. Основна частина роботи

						ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			29

ДП 6306.00.000 ПЗ

системи полягає саме у серверній частині. Перевагою для користувачів є те, що робота системи не залежить від машини клієнта, необхідна лише наявність одного з вищевказаних веб-браузерів для коректного відображення інтерфейсу і можливості використання усіх запропонованих функцій. Розробка не вимагає налаштування системи для роботи під різними операційними системами, наприклад, Windows, Linux, MacOS.

Розробка системи відбувається на локальній машині, після чого за допомогою системи контролю версій Git завантажується на веб-сервер та має змогу отримувати оновлення, уникаючи безпосередньої взаємодії з ОС веб-сервера.

Схема розгортання веб-застосунку показана на рисунку 4.1.

Рисунок 4.3 — Схема веб-розгортання системи

Після створення початкової версії системи, розробник завантажує її до репозиторію GitHub [13]. Наступним кроком є створення веб-сервера у системі DigitalOcean [14] та його налаштування: встановлення NGINX [15], який надає змогу системі працювати без вмикання її «вручну», а також надалі буде взаємодіяти з локальною машиною за допомогою gunicorn — Python WSGI HTTP сервера. Далі на сервер завантажується проект з репозиторію GitHub та виконується міграція бази даних з локальної машини.

4.3.1 Діаграма класів

В процесі розробки системи класифікації архетипу персонажа літературного твору використовувався як об'єктно-орієнтований підхід, так і функціональна парадигма програмування.

Об'єктно-орієнтоване програмування використовується для взаємодії системи з базою даних шляхом створення моделі, яка є аналогом

						ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			30

ДП 6306.00.000 ПЗ

Функціональне

представлення даних у базі MySQL. Кожна таблиця у базі даних відповідає класу у моделі.

програмування  
надає змогу  
розроблювати веб-

застосунки на базі мови програмування Python. Кожна окрема сторінка має функцію з декоратором-обробником, які реалізують функціонал застосунку, що є загальноприйнятим підходом при створенні систем за допомогою даної технології.

Для розробки програмного продукту було створено класи, які наведені у таблиці 4.1.

Таблиця 4.1 – Опис класів	
Клас	Опис
User	Клас, що описує користувача. Необхідний для функції авторизації.
Book	Клас, що описує літературний твір.
Archetype	Клас, що описує архетипи персонажів.
Person	Клас, що описує персонажа, для якого проводиться класифікація.

Опис атрибутів наведених класів зазначено у таблиці 4.2.

Таблиця 4.2 – Опис атрибутів класів			
Клас	Атрибут	Тип даних	Опис
User	id	int	Ідентифікатор

		г				запису	
							Арх.

ДП 6306.00.000 ПЗ

Змн. Арк.№ докум.

Підпис Дата

31



ДП 6306.00.000 ПЗ			
	username	string	Логін користувача
	password	string	Пароль користувача
Arc hetype	id	int	Ідентифікатор запису
	name	string	Назва архетипу
	person	relationship	Особливий тип даних для зв'язку між таблицями (аналог зовнішніх ключів)
Book	id	int	Ідентифікатор запису
	name	string	Назва твору
	author	string	Автор твору
Таблиця 4.2 (продовження)			
	person	relationship	Особливий тип даних для зв'язку між таблицями (аналог зовнішніх ключів)
Person	id	int	Ідентифікатор запису
ДП 6306.00.000 ПЗ			Арк.
Змн.	Арк. № докум.	Підпис	Дата
			32

ДП 6306.00.000 ПЗ			
	name	strin	Ім'я персонажа
	book_id	int	Зовнішній ключ до класу Book
	archetype_id	int	Зовнішній ключ до класу Archetype

Графічне зображення діаграми наведено у графічних матеріалах.

### 4.3.2 Діаграма послідовності

Для побудови діаграми послідовності необхідно розглянути послідовність дій користувача при взаємодії з системою.

Першим кроком є вхід користувача в систему та початок тренування моделі на підготованих датасетах. Якщо користувач не проводить тренування, використовується датасет за замовчуванням — українська проза. Це не є критичним, проте напряму впливає на точність класифікації моделі, адже якщо тренування проведене на даних з української літератури, почерк зарубіжних авторів може бути неправильно «зрозумілий» системою.

У разі правильної послідовності дій, користувач запускає тренування системи, запускається відповідний алгоритм, описаний вище, і система навчається класифікувати текст за наявними даними.

Наступним кроком є безпосередньо класифікація архетипу персонажа. Користувачеві необхідно додати файл з дискового носія комп'ютера з текстом твору з розширенням \*.txt, а також вказати ім'я персонажа, для якого визначатиметься архетип. У разі невірної формату файлу або незаповнених полів система повідомить користувача про помилку і запропонує ввести дані ще раз. У разі введення коректних даних система виконає класифікацію і виконає перехід користувача на сторінку результатів роботи.

						ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			33

ДП 6306.00.000 ПЗ

У результатах роботи вказується назва файлу, ім'я героя, його архетип, а також точність класифікації. Дані записуються до бази і можуть бути переглянуті в меню «Історія класифікацій».

Система має сторінки, які не є обов'язковими, проте надають корисну інформацію. Це сторінки «Архетипи» та «Історія». При переході на них користувачеві надається інформація про існуючі архетипи в літературі та історія класифікацій системи відповідно.

Для адміністратора надається можливість входу в систему для можливості редагування бази даних шляхом взаємодії з інтерфейсом. При натисненні на елемент меню «Увійти» користувач переходить на сторінку авторизації, де йому надається змога ввести логін та пароль, щоб увійти у систему. Для авторизованих користувачів кнопка «Увійти» змінюється на «Вийти».

Згідно з описаними діями користувача побудовано UML-діаграму послідовності, яка представлена у графічних матеріалах.

Графічне зображення діаграми наведено у графічних матеріалах.

#### 4.3.3 Діаграма компонентів

Система класифікації архетипу персонажа літературного твору поділяється на такі компоненти:

- Сервер MySQL — зберігання даних про архетипи, твори та результати класифікацій, проведених користувачами, а також авторизаційні дані користувачів;
- Parser — містить набір пакетів для перетворення датасету до цільової форми, що містить лише дані про потрібного персонажа;
- Classifier — містить набір пакетів для виконання математичної складової проекту;
- Flask Application — містить набір пакетів для реалізації серверної частини застосунку;

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34



ДП 6306.00.000 ПЗ

- Web-server — містить набір пакетів для розгортання веб-застосунку у мережі;
- UI — містить набір пакетів, що становлять клієнтську частину веб-застосунку.

Необхідно розглянути кожен компонент і встановити залежності між ними.

Графічне зображення діаграми наведено у графічних матеріалах.

Сервер MySQL відповідає за зберігання даних про такі сутності, як «Архетип», «Літературний твір», «Персонаж» та «Користувач». Містить усі необхідні функції, що застосовуються для проведення аналізу щодо попереднього використання програми і створення статистичних висновків про роботу системи. Зв'язок з програмним кодом відбувається за допомогою пакета SQLAlchemy, який є зовнішньою бібліотекою мови програмування Python.

Компонент Parser відповідає за обробку наданого системою тексту, незалежно від виконаної операції — тренування чи класифікація. Під обробкою мається на увазі виділення з тексту окремих персонажів, які зазначені користувачем для відкидання зайвих розділів твору і створення готового до класифікації тексту.

Компонент Classifier відповідає за математичне забезпечення програмного коду, тобто виконує усі необхідні дії для отримання результату класифікації.

В першу чергу виконується векторизація тексту — токенизація, нормалізація, стеммізація, лематизація та чистка. Наступним кроком є використання алгоритмів машинного навчання на обробленому тексті для виконання задачі класифікації — передбачення належності наданого персонажа до однієї з категорій, заданих у процесі тренування моделі.

Компонент Flask Application відповідає за серверну частину застосунку, тобто є «мостом» між математичною складовою програмного продукту та його

						ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			35

ДП 6306.00.000 ПЗ

веб-сторonoю. Проводить такі операції як авторизація користувача, перехід між сторінками, а також надає можливість шляхом взаємодії з інтерфейсом надавати можливість здійснювати тренування та класифікацію.

Компонент Web-Server відповідає за розгортання веб-застосунку та безперервну роботи системи у мережі. У якості веб-сервера обрано nginx, який зарекомендував себе як надійний HTTP-сервер і зворотній проксі-сервер. Основною його функціональністю є обслуговування статичних запитів, розподіл навантаження та відмовостійкість, модульність та підтримка uwsgi.

Компонент UI відповідає за взаємодію користувача із системою. Створено набір пакетів, кожен з яких надає певний функціонал на стороні клієнта.

#### 4.3.4 Специфікація функцій

Оскільки функціонал веб-застосунку розроблено в основному з використанням функціонального підходу, наявна велика кількість методів, що виконують дії над веб-інтерфейсом застосунку, а також алгоритмічною частиною. Специфікація функцій наведена у таблиці 4.3.

Таблиця 4.3 – Специфікація функцій

Назва функції	Аргументи	Опис аргументів	Опис функції
@app.route ('/')	/	Сторінки у системі	Декоратор функцій, що здійснює взаємодію коду з серверної частини з клієнтською

Змн.	Арк.	№ докум.	Підпис	Дата	ДП 6306.00.000 ПЗ	Арк.
						36

ДП 6306.00.000 ПЗ

logout()	—	—	Виконує вихід користувача з системи, переносить його на головну сторінку
load_user(user_id)	user_id	Ідентифікатор користувача	Завантажує користувача до системи
login()	—	—	Передхід користувача на сторінку авторизації

Таблиця 4.3 (продовження)

archetypes()	—	—	Перехід користувача на сторінку «Архетипи»
edit_book(id)	id	Ідентифікатор літературного твору	Зміна даних у таблиці та базі даних
delete_person(id)	id	Ідентифікатор персонажа	Видалення запису з таблиці та

ДП 6306.00.000 ПЗ

Арк. 37

ДП 6306.00.000 ПЗ			
history()	—	—	бази даних Перехід на сторінку «Історія»
hash_password(password)	password	Текстовий рядок — пароль користувача	Шифрування паролю для зберігання у базі даних
verify_password(stored_password, provided_password)	stored_password provide_d_password	Пароль з бази даних Пароль, введений користувачем	Перевірка правильного введення паролю
preprocess(dataset)	dataset	Текст літературного твору	Виділення у літературному творі персонажів та формування нових текстів
Таблиця 4.3 (закінчення)			
load_files(path, categories, encoding)	path	Шлях до текстового файлу у системі	Завантаження файлів та їх переформатування в об'єкт
	categories	Категорії класифікації	
	encoding	Кодування	
svm()	—	—	Виконання алгоритму опорних векторів
fit(data,	data	Переформатовані	Знаходження
ДП 6306.00.000 ПЗ			
Змн.	Арх.	№ докум.	Підпис Дата

ДП 6306.00.000 ПЗ				
target)		дані у вигляді об'єкта	паттернів у даних	
	target	Об'єкт-ціль, очікуваний результат класифікації		
predict()	—	—	Виконання передбачення, повертає те значення, яке бачить користувач в результаті класифікації	
vectorizer()	—	—	Виконання попередньої обробки тексту	

Висновок до розділу

У даному розділі описана архітектура системи, вказано засоби розробки, визначено вимоги до технічного забезпечення сервера та клієнта, описано архітектуру системи.

Наведено детальний опис UML-діаграм класів, послідовності та компонентів, а також класів та функцій програмного продукту.

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

Джерел на сторінці: 1-18, 24-25, 41

Сторінка 36 з 53

ДП 6306.00.000 ПЗ

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

У даному розділі наведено основні дії користувача у системі, результати виконаних дій, ознайомлення з інтерфейсом користувача за допомогою скриншотів виконання програми та текстового опису наявних функцій системи.

Метою користувача є отримання класифікації архетипу персонажа певного літературного твору, обраного самим користувачем, а також розширення знань про архетипи та їх практичного використання.

5.1.1 Головне меню

Головне меню, скриншот якого зображено на рисунку 5.1 представляє собою сторінку з коротко викладеною інформацією про функціонал системи. Згору розташоване фіксоване меню навігації по системі, яке включає в себе такі сторінки як «Тренування», «Класифікація», «Архетипи», «Історія» та «Увійти». Також користувач має можливість одразу перейти до тренування системи, скориставшись кнопкою «Почати роботу», що знаходиться в основній частині меню.

Рисунок 5.1 — Головна сторінка програми

Кнопка меню «Увійти» призначена адміністраторові для авторизації у системі. Звичайному користувачеві не потрібно бути авторизованим для використання системи.

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

ДП 6306.00.000 ПЗ				
<p>5.1.2 Авторизація</p> <p>Після натиснення кнопки «Увійти» відкривається сторінка авторизації, яка зображена на рисунку 5.2.</p> <p>Рисунок 5.2 — Сторінка авторизації</p> <p>На даній сторінці розташовано такі елементи, як два текстових поля для введення логіна та пароля користувача, а також кнопка увійти, натискання якої перевіряє логін та пароль та авторизує адміністратора у системі.</p> <p>Для початку роботи з системою необхідно спочатку натиснути кнопку «Почати роботу» у головному меню, або скористатись меню вгорі і обрати елемент «Тренування». Це зумовлено тим, що задача класифікації потребує попереднього тренування моделі для виконання поставленої мети.</p> <p>5.1.3 Тренування</p> <p>Після натиснення елемента меню «Тренування» здійснюється перехід на сторінку, яка зображена на рисунку 5.3.</p> <p>Рисунок 5.3 — Сторінка тренування системи</p> <p>На даній сторінці користувач має можливість обрати один з датасетів, наявних у системі для підвищення точності класифікації, в залежності від очікуваного результату. Сторінка містить випадające меню з датасетами, які включають українську та зарубіжну прозу, а також кнопку «Тренування», яка безпосередньо виконає функцію тренування моделі у системі.</p>				
				ДП 6306.00.000 ПЗ
Змн.	Арк.	№ докум.	Підпис	Дата
				41

ДП 6306.00.000 ПЗ

Після виконання тренування моделі, користувачеві необхідно перейти до меню класифікація, скориставшись меню згору та обравши елемент «Класифікація».

5.1.4 Класифікація

Після натиснення кнопки «Класифікація» відбувається перехід до меню класифікації, що зображене на рисунках 5.4 та 5.5. На сторінці розміщені такі кнопки, як «Обрати файл» та «Визначити», а також текстове поле «Ім'я» для введення імені персонажа, для якого проводиться класифікація.

Рисунок 5.4 — Сторінка класифікації

Рисунок 5.5 — Сторінка класифікації

Користувач має можливість завантажити файл — текстовий файл, що містить літературний твір — з власного дискового носія для проведення класифікації на його основі. У текстовому поясненні вказано, що файл мусить мати розширення \*.txt, інакше виконання класифікації не відбуватиметься. Також необхідно вказати ім'я героя, для якого визначатиметься архетип в обраному літературному творі. Після заповнення всіх полів необхідно натиснути кнопку «Виконати» для здійснення класифікації. Дана дія завантажує сторінку результатів класифікації.

5.1.5 Результати класифікації

Результати класифікації зображені на рисунках 5.5 та 5.6 для української та зарубіжної літератури відповідно. Сторінка результатів має текстовий опис виконаної класифікації архетипу, а саме файл, з якого узято

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42



ДП 6306.00.000 ПЗ

текст літературного твору, ім'я класифікованого персонажа, його визначений архетип, точність класифікації і кнопка «Повернутись».

Рисунок 5.6 — Сторінка результатів класифікації для української прози

Рисунок 5.7 — Сторінка результатів класифікації для української прози

На скриншоті бачимо виконану класифікацію архетипу відповідно до даних, введених у систему, що зображені на рисунках 5.4 та 5.5 відповідно. При натисненні кнопки «Повернутись» завантажується сторінка «Класифікація», що описана вище.

5.1.6 Архетипи

Користувач має можливість обрати елемент меню «Архетипи» для перегляду короткого опису кожного з основних архетипів у літературі. Сторінка «Архетипи» зображена на рисунках 5.8 та 5.9. Дана сторінка не має активних елементів, таких як кнопки, вона є описовою, створена з метою надати базову характеристику використаних у системі архетипів.

Рисунок 5.8 — Сторінка описової характеристики архетипів (1 частина)

Рисунок 5.9 — Сторінка описової характеристики архетипів (2 частина)

					ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

ДП 6306.00.000 ПЗ

5.1.7 Історія класифікацій

Користувач має можливість обрати елемент меню «Історія» для перегляду історії використання системи. Сторінка «Історія» для звичайного користувача зображена на рисунку 5.10.

Рисунок 5.10 — Сторінка історії використання системи (користувач)

На сторінці історії класифікацій зображено таблицю, яка містить автора твору, назву, героя, для якого проведена класифікація, визначений архетип, а також точність класифікації моделі.

Для адміністратора сторінка історії має додатковий функціонал, який надає змогу йому редагувати записи у базі даних, такі як назва твору та автор, а також видалення запису. Сторінку «Історія класифікацій» для адміністратора показано на рисунку 5.11.

Рисунок 5.11 — Сторінка історії використання системи (адміністратор)

На сторінці історії використання системи адміністратор має дві кнопки навпроти кожного запису, які дозволяють редагувати або видаляти записи. Меню редагування назви твору та автора зображено на рисунку 5.12.

Рисунок 5.12 — Сторінка редагування запису (адміністратор)

ДП 6306.00.000 ПЗ

Арк.

44

Змн. Арк. № докум. Підпис Дата

Джерел на сторінці: 1-18, 20-21, 25

Сторінка 41 з 53

ДП 6306.00.000 ПЗ

## 5.2 Випробування програмного продукту

### 5.2.1 Мета випробувань

Метою випробувань являється перевірка відповідності функцій комплексу задач класифікації архетипу персонажа у літературному творі вимогам технічного завдання.

### 5.2.23 загальні положення

Випробування проводяться на основі наступних документів:

ГОСТ 34.603 92. Інформаційна технологія. Види випробувань автоматизованих систем;

ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

### 5.2.3 Результати випробувань

У даному підрозділі перевірено основний функціонал системи та відповідність роботи системи до поставлених функціональних вимог. Перелік випробувань наведено у таблицях 5.1-5.7.

Таблиця 5.1 – Тестування роботи тренування

Мета тесту	Перевірка функції «Тренування»
Початковий стан системи	Відкрито меню «Тренування»
Вхідні дані	Датасети у системі
Схема проведення тесту	Вибір датасету з випадального меню та натискання на кнопку «Тренування» для проведення тренування моделі
Очікуваний результат	Успішне тренування моделі на основі обраного датасету
Стан системи після	Успішне тренування моделі на основі обраного

Змн. Арк. № докум. Підпис Дата

ДП 6306.00.000 ПЗ

45



ДП 6306.00.000 ПЗ				
Мета тесту проведення випробування		Перевірка функції «Тренування» датасету		
Таблиця 5.2 – Тестування роботи класифікації				
Мета тесту		Перевірка роботи меню «Класифікація»		
Початковий стан системи		Відкрито меню «Класифікація»		
Вхідні дані		Текстовий файл користувача та введене ім'я персонажа у текстове поле		
Таблиця 5.2 (продовження)				
Схема проведення тесту		Вибір файлу з дискового носія комп'ютера користувача, введення імені персонажа та натискання кнопки «Визначити»		
Очікуваний результат		Успішна класифікація та перехід на сторінку результатів класифікації		
Стан системи після проведення випробування		Успішна класифікація та перехід на сторінку результатів класифікації		
Таблиця 5.3 – Тестування виведення результатів класифікації				
Мета тесту		Перевірка функції «Результати»		
Початковий стан системи		Відкрито меню «Результати»		
Вхідні дані		Результати класифікації		
Схема проведення тесту		Натискання кнопки «Визначити» у меню класифікації		
Очікуваний результат		Виведення назви файлу, імені персонажа, архетипу та точності класифікації моделі		
Стан системи після проведення		Виведення назви файлу, імені персонажа, архетипу та точності класифікації моделі		
ДП 6306.00.000 ПЗ				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата
				46

ДП 6306.00.000 ПЗ					
Мета тесту випробування	Перевірка функції «Результати»				
Таблиця 5.4 – Тестування виведення історії класифікацій					
Мета тесту	Перевірка функції «Історія класифікацій»				
Початковий стан системи	Відкрито меню «Історія класифікацій»				
Таблиця 5.4 (продовження)					
Вхідні дані	Дані з бази про проведену класифікацію: автор та назва твору, ім'я персонажа, архетип, точність класифікації моделі				
Схема проведення тесту	Натискання кнопки «Історія» у головному меню				
Очікуваний результат	Виведення автора, назви твору, імені персонажа, архетипу та точності класифікації моделі у таблицю				
Стан системи після проведення випробування	Виведення автора, назви твору, імені персонажа, архетипу та точності класифікації моделі у таблицю				
Таблиця 5.5 – Тестування авторизації					
Мета тесту	Перевірка функції «Авторизація»				
Початковий стан системи	Відкрито меню «Авторизація»				
Вхідні дані	Логін та пароль, введені в текстове поле користувачем				
Схема проведення тесту	Натискання кнопки «Увійти»				
Очікуваний результат	Користувача авторизовано і перенаправлено до сторінки головного меню				
Стан системи після проведення	Користувача авторизовано і перенаправлено до сторінки головного меню				
ДП 6306.00.000 ПЗ					
Змн.	Арк.	№ докум.	Підпис	Дата	Арк.
					47

ДП 6306.00.000 ПЗ	
Мета тесту випробування	Перевірка функції «Авторизація»
Таблиця 5.6 – Тестування видалення запису адміністратором	
Мета тесту	Перевірка функції «Видалити»
Початковий стан системи	Відкрито меню «Історія класифікацій»
Вхідні дані	Користувач авторизований як адміністратор та має доступ до кнопок «Редагувати» та «Видалити»
Схема проведення тесту	Натискання кнопки «Видалити»
Очікуваний результат	Запис видалено з таблиці і бази даних
Стан системи після проведення випробування	Запис видалено з таблиці і бази даних
Таблиця 5.7 – Тестування редагування запису адміністратором	
Мета тесту	Перевірка функції «Редагування»
Початковий стан системи	Відкрито меню «Історія класифікацій»
Вхідні дані	Користувач авторизований як адміністратор та має доступ до кнопок «Редагувати» та «Видалити»
Схема проведення тесту	Натискання кнопки «Редагувати», після появи спливаючого вікна редагування даних та натискання кнопки «Зберегти»
Очікуваний результат	Запис змінюється в таблиці і у базі даних
Стан системи після проведення випробування	Запис змінюється в таблиці і у базі даних
ДП 6306.00.000 ПЗ	
Змн.	Арк.
№ докум.	Підпис
Дата	48

ДП 6306.00.000 ПЗ				
Висновок до розділу				
У даному розділі описано керівництво користувача на прикладі навігації системою та графічного зображення роботи системи. Наведено основну діяльність користувача та результати певних дій. Проведено тестування функціональних вимог методом випробувань, результати описано у відповідних таблицях.				



ДП 6306.00.000 ПЗ

## ЗАГАЛЬНІ ВИСНОВКИ

У дипломному проекті реалізовано інформаційно-аналітичну систему класифікації архетипу персонажа. Метою системи є спрощення процесу літературного аналізу творів та дослідження залежності архетипів персонажів літературних творів від певних факторів. Задачами розробки є тренування моделі на різних вибірках даних, визначення архетипу введеного персонажа у певному творі, отримання описової характеристики кожного архетипу, отримання історії використання програми. Користувач системи матиме можливість провести класифікацію власного тексту для визначення архетипу обраного персонажа.

У розділі загальних положень описано основні цілі класифікації тексту як напряму у сфері машинного навчання і визначено необхідність розробки поглибленого аналізу літературних творів як для розвитку сфери в цілому, так і для прикладного застосування системи.

У розділі інформаційного забезпечення виконано планування та побудова структури бази даних, описано сутності у вигляді таблиць, наведено графічне зображення структури бази даних.

У розділі математичного забезпечення описано змістовну та математичну постановку, наведено алгоритм розв'язання поставленої задачі.

У розділі програмного та технічного забезпечення описана архітектура системи, вказано засоби розробки, визначено вимоги до технічного забезпечення сервера та клієнта, описано архітектуру системи. Наведено детальний опис класів та функцій програмного продукту, побудовано UML-діаграми.

У технологічному розділі описано керівництво користувача на прикладі навігації системою та графічного зображення роботи системи. Проведено тестування функціональних вимог методом випробувань.

						ДП 6306.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			50

ДП 6306.00.000 ПЗ				



## Схожість

Схожість із джерелами з Інтернету

15

20	<a href="https://ela.kpi.ua/bitstream/123456789/31069/1/Yashchuk_bakalavr.pdf">https://ela.kpi.ua/bitstream/123456789/31069/1/Yashchuk_bakalavr.pdf</a>	1.98%
21	<a href="https://kxtp.kpi.ua/theses/2018-zhezherun.pdf">https://kxtp.kpi.ua/theses/2018-zhezherun.pdf</a>	1.24%
26	<a href="https://refdb.ru/look/2533440-p5.html">https://refdb.ru/look/2533440-p5.html</a>	0.65%
29	<a href="https://ela.kpi.ua/bitstream/123456789/30983/1/Ngo_bakalavr.pdf">https://ela.kpi.ua/bitstream/123456789/30983/1/Ngo_bakalavr.pdf</a>	0.57%
39	<a href="https://kxtp.kpi.ua/theses/kulyk-bachelor.pdf">https://kxtp.kpi.ua/theses/kulyk-bachelor.pdf</a>	0.34%
42	<a href="https://ela.kpi.ua/bitstream/123456789/29397/1/Andreieva_bakalavr.docx">https://ela.kpi.ua/bitstream/123456789/29397/1/Andreieva_bakalavr.docx</a>	0.3%
43	<a href="https://ela.kpi.ua/bitstream/123456789/29384/1/Shulikov_bakalavr.pdf">https://ela.kpi.ua/bitstream/123456789/29384/1/Shulikov_bakalavr.pdf</a>	0.3%
55	<a href="https://ukrbukva.net/page,7,92577-Metodika-razrabotki-programmnogo-produkta-dlya-poiska-prichin-v-izmeneniyah-trendov-v...">https://ukrbukva.net/page,7,92577-Metodika-razrabotki-programmnogo-produkta-dlya-poiska-prichin-v-izmeneniyah-trendov-v...</a>	0.13%

Схожість по Бібліотеці акаунту

162

1	Yablonskyi_bachelor	ID файлу: 5839560	Institution: National Technical University of Ukraine "Kyiv Polytechnic Instit...	3.26%
2	Serduk_bachelor	ID файлу: 5824670	Institution: National Technical University of Ukraine "Kyiv PolytechnicInstitute"	3.26%
3	Portjanij_bachelor_is63	ID файлу: 1003896804	Institution: National Technical University of Ukraine "Kyiv Polytechnic...	3.22%
4	Medvednikov_bachelor	ID файлу: 1000017739	Institution: National Technical University of Ukraine "Kyiv Polytechnic...	3.14%
5	Potilchak_bachelor	ID файлу: 12272972	Institution: National Technical University of Ukraine "Kyiv Polytechnic Instit...	3.19%
6	Kotkova_bachelor	ID файлу: 12272924	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institu...	3.15%
7	Rjabcev_bachelor	ID файлу: 12272971	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institu...	3.14%
8	Kaminskij_bachelor	ID файлу: 1000044112	Institution: National Technical University of Ukraine "Kyiv Polytechnic Ins...	3.01%
9	Polupan_bachelor	ID файлу: 5839485	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	3.09%
10	Kazmirchuk_bachelor	ID файлу: 12191135	Institution: National Technical University of Ukraine "Kyiv Polytechnic Ins...	2.98%
11	Homenko_bachelor_is61	ID файлу: 1003897031	Institution: National Technical University of Ukraine "Kyiv Polytechn...	2.95%

12	Didik_Barishich_bachelor	ID файлу: 5849901	Institution: National Technical University of Ukraine "Kyiv Polytechnic I...	2.9%
13	Lahus_Dmytro	ID файлу: 5882636	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	3%
14	Madojan_Shishman_bachelor	ID файлу: 1000017976	Institution: National Technical University of Ukraine "Kyiv Polyt...	2.98%
15	Kolisetsky_bachelor_IP43	ID файлу: 1000070098	Institution: National Technical University of Ukraine "Kyiv Polytechn...	2.85%
16	Stek_bachelor_ISZ41	ID файлу: 5882677	Institution: National Technical University of Ukraine "Kyiv Polytechnic Instit...	2.67%
17	Fomin_bachelor	ID файлу: 1003798192	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institu... 2 Джерело	2.77%
18	Dubina_Martinuk_bachelor	ID файлу: 12272929	Institution: National Technical University of Ukraine "Kyiv Polytechn...	2.77%
19	Borisenko_bachelor	ID файлу: 1000052293	Institution: National Technical University of Ukraine "Kyiv Polytechnic In...	2.32%
22	Mosaichuk	ID файлу: 1000762626	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	0.92%
23	Lubivij_bachelor	ID файлу: 1000044105	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institu...	0.79%
24	Barishich_bachelor	ID файлу: 5849636	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institu...	0.75%
25	Klimenko_bachelor	ID файлу: 5824711	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institu...	0.69%
27	Butusov_bachelor	ID файлу: 5857457	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	0.62%
28	Maksimenko_bachelor	ID файлу: 5910322	Institution: National Technical University of Ukraine "Kyiv Polytechnic Ins...	0.49%
30	Klimenko_Voloshin_bachelor	ID файлу: 5824717	Institution: National Technical University of Ukraine "Kyiv Polytechn...	0.54%
31	Doroshenko_Kochubej_bachelor (1)	ID файлу: 1000018093	Institution: National Technical University of Ukraine "Kyiv...	0.49%
32	Симотюк_MB_КВ-51_бак_пр_2019	ID файлу: 1000072049	Institution: National Technical University of Ukraine "Kyiv...	0.48%
33	Гребенюк КА-74мп	ID файлу: 8081528	Institution: National Technical University of Ukraine "Kyiv Polytechnic Insti...	0.48%
34	Zavads kij_bachelor	ID файлу: 5910455	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institu...	0.35%
35	Talko_magistr_ZIS	ID файлу: 8241219	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	0.47%
36	Студентська робота	ID файлу: 3480095	Institution: Lviv Polytechnic National University	0.42%
37	Bondar_bachelor	ID файлу: 5882627	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	0.35%

38	Tsaturjan_bachelor	ID файлу: 5882688	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institu...	0.24%
40	Chernenko_bachelor	ID файлу: 5938239	Institution: National Technical University of Ukraine "Kyiv Polytechnic Instit...	0.23%
41	Javorskij_bachelor_is63	ID файлу: 1003896996	Institution: National Technical University of Ukraine "Kyiv Polytechnic...	0.34%
44	Galaktionov_bachelor	ID файлу: 5882634	Institution: National Technical University of Ukraine "Kyiv Polytechnic Inst...	0.13%
45	Bichkovskaja(Shevchenko)_bachelor_isz51	ID файлу: 1000092581	Institution: National Technical University of Ukrain...	0.24%
46	Perevertailo_bachelor_IS43	ID файлу: 6002527	Institution: National Technical University of Ukraine "Kyiv Polytechni...	0.23%
47	Kasjan_bachelor_ISz41	ID файлу: 5978351	Institution: National Technical University of Ukraine "Kyiv Polytechnic Ins...	0.23%
48	МД_Золкіна В.О._БС-81мп	ID файлу: 1000807345	Institution: National Technical University of Ukraine "Kyiv Polyte...	0.17%
49	ПЗБП_АУТС_КПІ_2018_СагайдаковДО	ID файлу: 6005162	Institution: National Technical University of Ukraine "K...	0.16%
50	Kit_Dmytro_IK-51	ID файлу: 1000057309	Institution: National Technical University of Ukraine "Kyiv Polytechnic Instit...	0.16%
51	Романчук-окрема-частина	ID файлу: 1000048400	Institution: National Technical University of Ukraine "Kyiv Poly...	0.16%
52	Кириченко_all	ID файлу: 8416494	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	0.14%
53	ПЗ_Мисік_Артем_БС-43	ID файлу: 5987903	Institution: National Technical University of Ukraine "Kyiv Polytechnic ...	0.13%
54	Диплом_Паляничко	ID файлу: 1000789712	Institution: National Technical University of Ukraine "Kyiv Polytechnic...	0.13%
56	Lubchenko_Romanchenko_bachelor	ID файлу: 5857484	Institution: National Technical University of Ukraine "Kyiv Po...	0.13%
57	Студентська робота	ID файлу: 1000048834	Institution: Lviv Polytechnic National University	0.11%
58	Zinchenko_bachelor	ID файлу: 5850166	Institution: National Technical University of Ukraine "Kyiv Polytechnic Instit...	0.11%

## Цитати

Цитати	2
--------	---

- 1

4.2 Вимоги до технічного забезпечення 4.2.1 Загальні вимоги Для правильної роботи системи класифікації та тренування моделі необхідний сервер з наступними мінімальними параметрами: — процесор з частотою 1,6 ГГц або вище; — об'єм оперативної пам'яті 2 ГБ або більше; — жорсткий диск об'ємом не менше 20 ГБ; — операційна система Linux (Fedora 31, Ubuntu 16.04 або 18.04).
- 2

«Війна і мир»

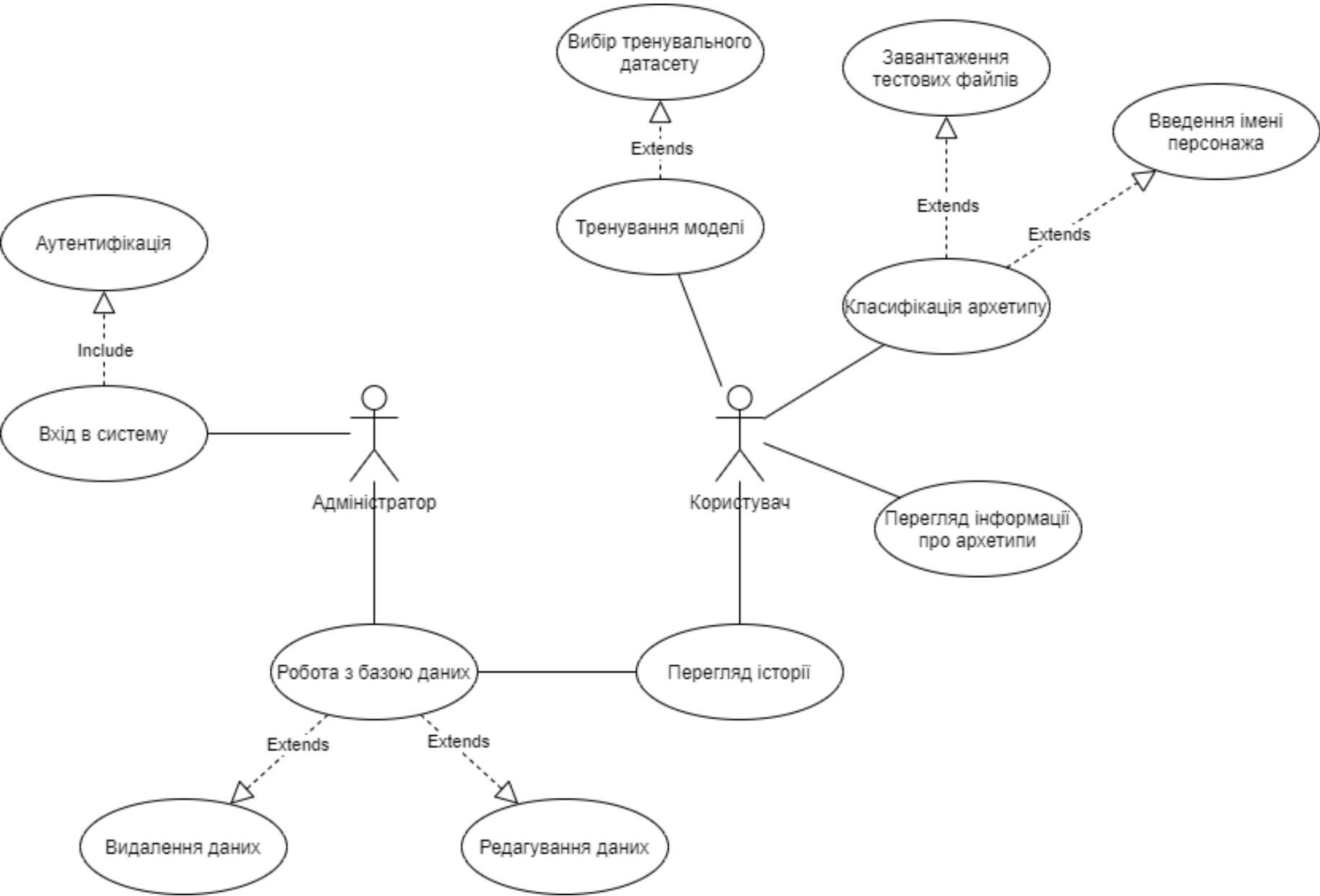
# **Графічний матеріал до дипломного проєкту**

на тему: Інформаційно-аналітична система класифікації архетипу  
персонажа

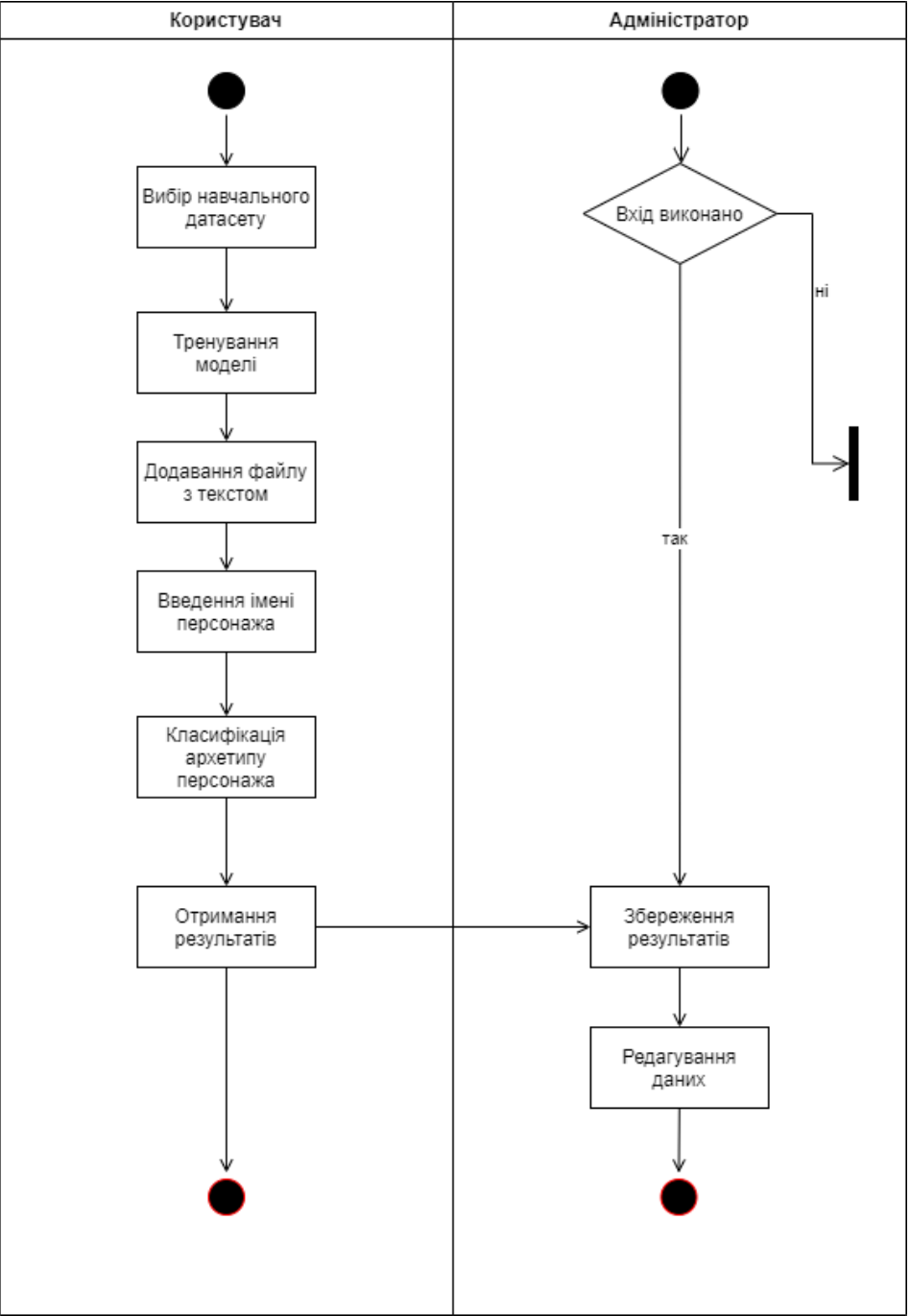
---

Київ – 2020 року

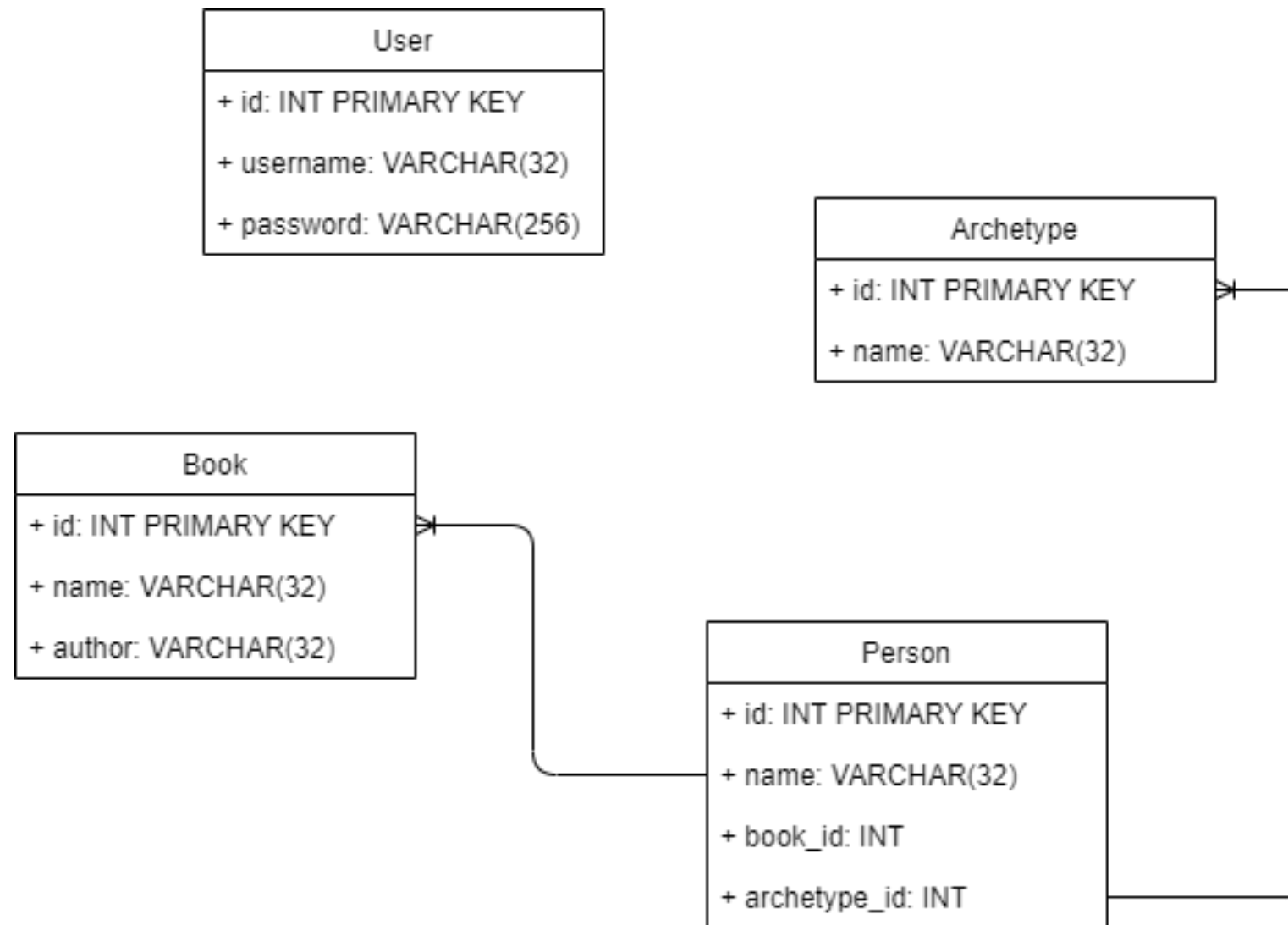




					ДП 6306.02.000 ССВ								
					Схема структурна варіантів використань				Літера		Маса	Масштаб	
									Аркуш 1		Аркушів 1		
Зм.	Арк.	№ документа	Підпис	Дата	Інформаційно-аналітична система класифікації архетипу персонажа				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63				
Розробив	Восієв Р.З.												
Перевішив	Фіногенов О.Д.												
Консульт.													
Н. кон.	Проскура С.Л.												
Затвердив	Павлов О.А.												

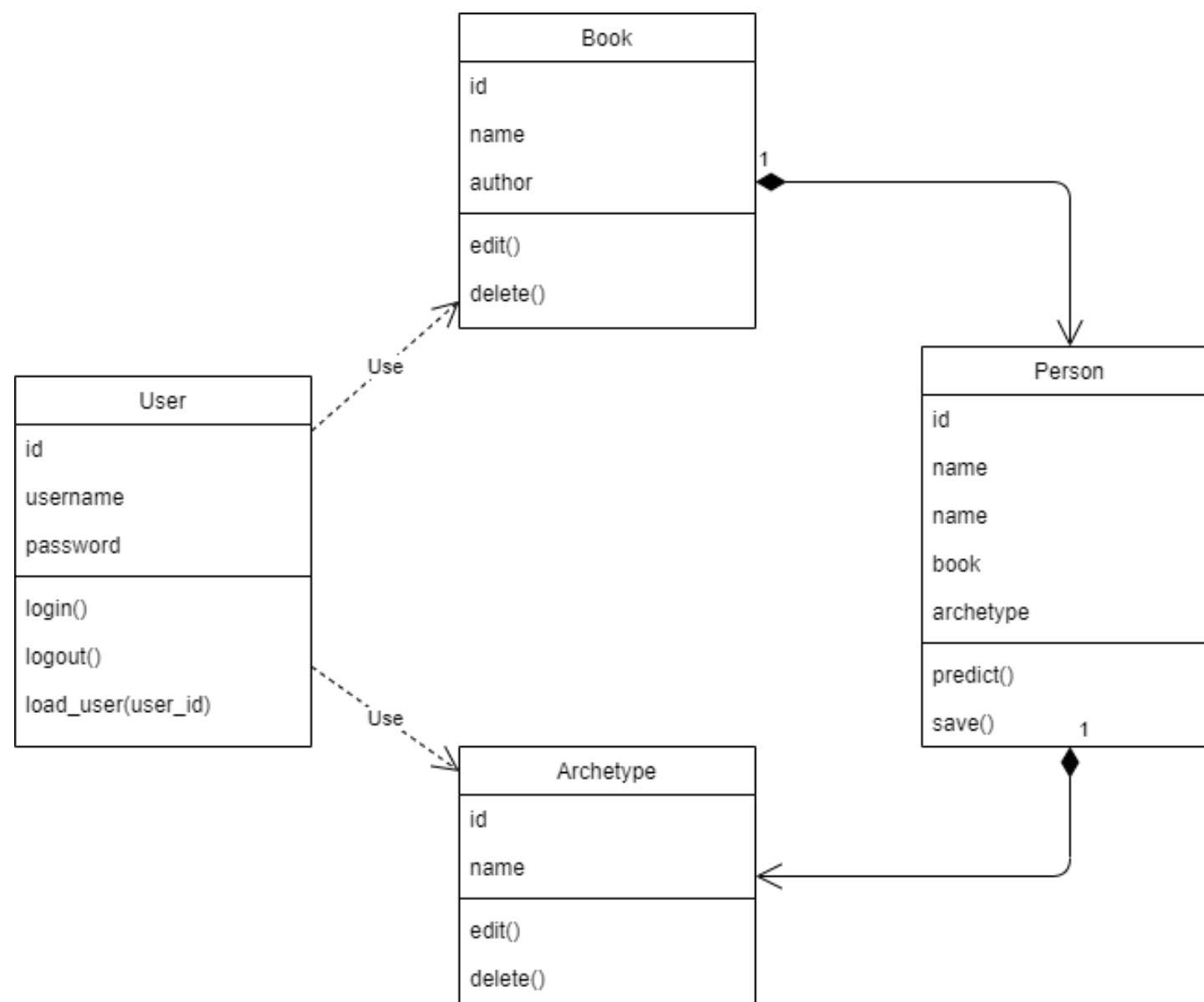


					ДП 6306.03.000 ССД			
					Схема структурна діяльності	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив		Восієв Р.З.			Інформаційно-аналітична система класифікації архетипу персонажа	Аркуш 1		Аркушів 1
Перевірив		Фіногенов О.Д.				КПІ ім. Ігоря Сікорського		
Консульт.						Каф. АСОІУ		
Н. кон.		Проскура С.Л.				Гр. ІС-63		
Затвердив		Павлов О.А.						

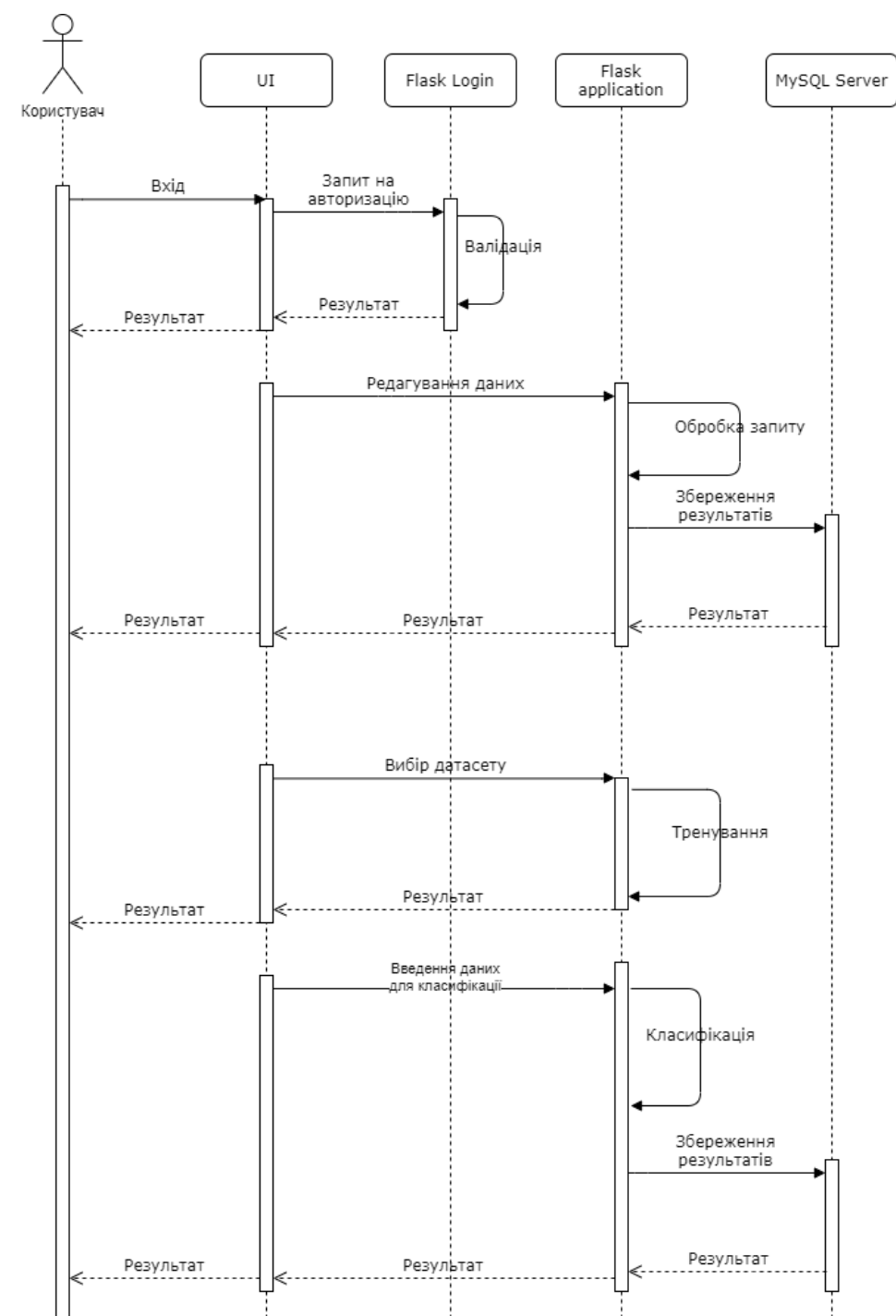


					ДП 6306.04.000 СБД							
Зм.	Арк.	№ документа	Підпис	Дата	Схема бази даних			Літера		Маса	Масштаб	
Розробив	Восієв Р.З.											
Перевірів	Фіногенов О.Д.							Аркуш 1		Аркушів 1		
Консульт.								Інформаційно-аналітична система класифікації архетипу персонажа				
Н. кон.	Проскура С.Л.											
Затвердив	Павлов О.А.							КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63				



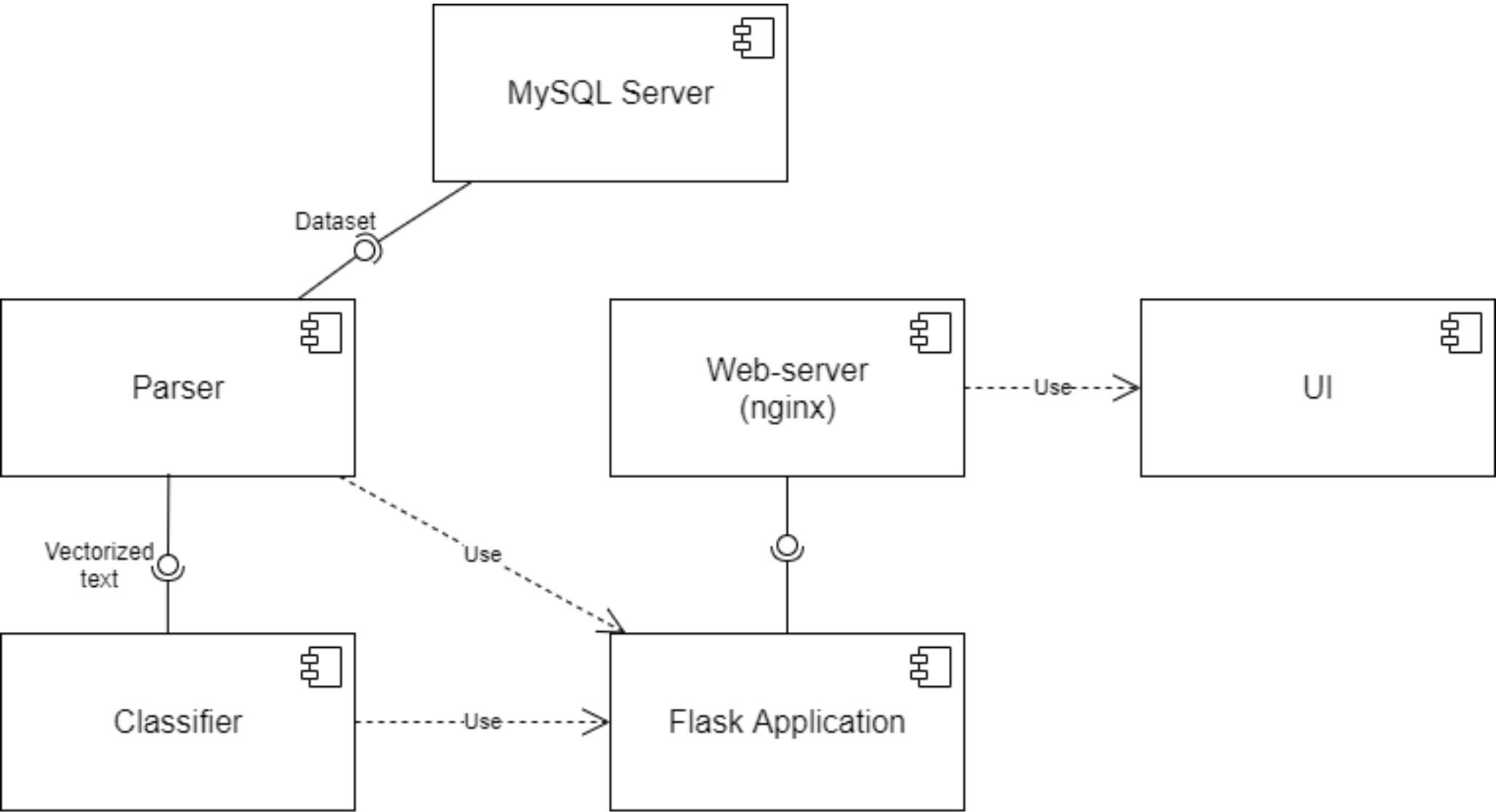


					ДП 6306.05.000 ССК				
					Схема структурна класів				
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Восієв Р.З.							
Перевірів		Фіногенов О.Д.							
Консульт.									
Н. кон.		Проскура С.Л.							
Затвердив		Павлов О.А.							
					Інформаційно-аналітична система класифікації архетипу персонажа				
					Літера    Маса    Масштаб				
					Аркуш 1    Аркушів 1				
					КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63				



					ДП 6306.06.000 ССП				
					Схема структурна послідовності				
Зм.	Арк.	№ документа	Підпис	Дата	Інформаційно-аналітична система класифікації архетипу персонажа				
Розробив		Восієв Р.З.							
Перевірів		Фіногенов О.Д.							
Консульт.					Інформаційно-аналітична система класифікації архетипу персонажа				
Н. кон.		Проскура С.Л.							
Затвердив		Павлов О.А.			КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63				





					ДП 6306.07.000 ССК									
					Схема структурна компонентів					Літера		Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата										
Розробив	Восієв Р.З.													
Перевірив	Фіногенов О.Д.									Аркуш 1		Аркушів 1		
Консульт.					Інформаційно-аналітична система класифікації архетипу персонажа					КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63				
Н. кон.	Проскура С.Л.													
Затвердив	Павлов О.А.													